ASRS-CMFS: Using a custom Transformer-based model to predict anomalies in aviation incident reports

Samuel Kierszbaum, Thierry Klein, Laurent Lapasset

ASRS-CMFS: Using a custom Transformer-based model to predict anomalies in aviation incident reports

Samuel Kierszbaum[1*], Thierry Klein[2], Laurent Lapasset[3]

[1] ENAC - Ecole Nationale de l'Aviation Civile , Université de Toulouse, Enac. samuel.kierszbaum@enac.fr
[2]Institut de Mathématiques de Toulouse; UMR5219. Université de Toulouse; ENAC - Ecole Nationale de l'Aviation Civile , Université de Toulouse, France. thierry01.klein@enac.fr
[3] ENAC - Ecole Nationale de l'Aviation Civile , Université de Toulouse. laurent.lapasset@recherche.enac.fr

*Corresponding author: Samuel Kierzbaum, kierszbaumsamuel@gmail.com

December 17, 2021

Abstract

In this article, the authors built and used a custom transformer-based model, based on a compact version of RoBERTa, named ASRS-CMFS, to classify aviation incident reports. The classification is applied to fourteen distinct sets of specific aviation incident-related anomalies, such as Aircraft Equipment problems or Altitude Deviation problems. The authors extracted the incident reports and the associated fourteen sets of categories from the Aviation Safety Reporting System.
After discussing the choice of evaluation metric, the authors evaluated the model using the Matthews Correlation Coefficient metric. To measure the precision of the scores obtained on the different text classification problems, the authors provided the results with confidence intervals. They also used statistical hypothesis testing to evaluate the impact of the document length on the performance of the custom model. The authors provided a mathematical demonstration for the use of confidence intervals and hypotheses testing on MCC values. Finally, the authors discussed whether the model was fit for use in a professional environment.
The authors found that while the model showed promising results, this question could only remain unanswered at this stage, but the steps to take are clear. The authors also proposed that hypothesis testing could

be valuable in any situation where one wanted to study the impact of a particular document feature on the performance of a document classifier.

Keywords— ASRS, BERT, MCC, NLP, aviation

1 Introduction

With the growth of global air traffic and the development of "just culture" (Pellegrino, 2019) within the different aviation institutions and actors, an increasing quantity of incidents is reported. In the framework of aviation safety, this phenomenon spurs the need for some form of support to process the reports, such as automatic document classification (Tulechki, 2015; Zhang u. a., 2021; Darveau u. a., 2020; Yang, 2020). In this article, we work on approximately 40 000 occurrences extracted from the NASA's semi-structured ASRS dataset (Aviation Safety Reporting System). Created in 1976, it contains voluntarily and anonymized reported descriptions of incident occurrences in an aviation context (ASRS, 2019), as well as metadata. We obtained our version of the public dataset through a request on the ASRS website and received it on a disk. For information, an extensive description of the dataset is available in Appendix B.
We aim to classify the reported aviation occurrence narratives on 14 different text classification (TC) problems. Each of the TC problems is made of categories that can be found in ASRS aviation incidents under the form of metadata, describing an anomaly such as aircraft equipment related problems or altitude deviation problems (for instance "undershoot" or "overshoot"). The class composition of each of these classification problems are given in Tables 4 and 8. Text mining ASRS incidents has been done using various natural Language Processing (NLP) text classification or topic modeling techniques such as SVM (Tanguy u. a., 2016), LDA (Robinson, 2019), RNNs (Yang, 2020; Howard und Ruder, 2018) among other techniques.
In our case, we decided to use the recent transformer-based technology (Vaswani u. a., 2017) on which the current state-of-the-art models in the Natural Language Understanding landscape (NLU) are based (Wang u. a., 2019, 2020), such as BERT (Devlin u. a., 2019) or the more recent T5 model (Raffel u. a., 2020).
In particular, we built, pre-trained from scratch, and fine tuned a custom transformer-based model, based on a compact version of the classical RoBERTa model (Liu u. a., 2019). Once the model was built, we assessed its performance on the classification task, using the Matthews correlation coefficient (MCC) score (Chicco und Jurman, 2020). We provided results with confidence intervals and proposed a tool to investigate if a particular document feature impacted the model's performance. Finally, we discussed the viability of using the model to support ASRS analysts in the field.
The article is organized as follows: In Section 2, we describe and explain our choices for the architecture of our custom model, the pre-training strategy, the pre-training, and fine tuning data and parameters.
In Section 3, we define our evaluation metric and motivate using it in particular. We also provide the theorems that we constructed to obtain confidence intervals for our results and show how hypothesis testing can be used to get insightful information on our model's performance regarding the input documents' features.
In Section 4, we show and comment our results.
Finally, in Section 5, we discuss the usefulness of hypothesis testing, as well as the viability of our model for field use.

2 Pre-training and fine tuning of the model

2.1 Pre-training

2.1.1 Custom model's architecture and pre-training parameters

In this section, we describe our model's architecture. The model we work with is transformer-based. We provide a general introduction to such models in Appendix C for information.
Aside from the difference in size, our model functions similarly to the more classical RoBERTa (Liu u. a., 2019). We chose RoBERTa as our base because it is a well-studied and sturdy transformer-based model with good performance in the Natural Language Understanding landscape (Wang u. a., 2019, 2020).
Regarding the size of our model, we know that bigger models tend to be more sample-efficient than smaller models (Kaplan u. a., 2020), meaning that they outperform smaller models with fewer training steps. However, we don't know if these results remain true in extreme cases of small pre-training data volume such as ours. In general, bigger models need more computation steps to attain peak performances, which in our case would lead to a high number of repetitions of the data. This can negatively impact our learning (Raffel u. a., 2020).
In contrast, compact models require small pre-training data and time required to attain peak performance (Micheli u. a., 2020). We initially supposed that the gains from using a smaller model in our context of work out-weighted the potential adverse effects. Aside from these considerations, the pros of using a compact model include low latency, less memory consumption, less computing resources requirements. Even in this context, our model pre-trained for 14 days on the 8 total GPUs available in our lab (see Appendix A for details on hardware), using Pytorch for parallelization (Paszke u. a., 2019). All information related to the pre-training including the training loss curve can be found in Kierszbaum (2021).
Our custom model's architecture and pre-training parameters are described in Table 1. RoBERTa's architecture and pre-training parameters are also given for comparison.

Hyperparam	Custom	RoBERTa
Layers	12	12
Hidden dimension	256	768
FFN inner hidden size	1024	3072
Attention Head	4	12
Learning rate	5e-4	6e-4
Train batch size	64	8k
Gradient accumulation steps	2	-
Weight Decay	0.01	0.01
# of parameters	18M	125M
pre-training steps	800k	500k
Tokenizer	custom	general purpose
Vocab size	32005	50265

Table 1: Comparison of architecture and hyperparameters for pre-training

As seen in Table 1, when building the model, we favored maintaining depth (number of layers) while reducing width (number of elements in a single layer: hidden dimension size, FFN inner hidden size, number of attention heads) to maintain maximum efficiency, following the work of Turc u. a. (2020). We used Electra-small (Clark u. a., 2020) as our reference for our pre-training parameters such as the learning rate, batch size, and weight decay because the size of the two models is comparable: 18M for our model, 14M for Electra-small.

2.1.2 Pre-training strategy

In this Section, we explain that there are mostly two pre-training strategies when using a language model on domain-specific data, and provide justification for our choice of strategy.

The textual data we are working with has been produced by people in aviation, reporting on aviation incidents in English. For this reason, the language used is intrinsically vastly different than general English. This feature of our data is referred to as "in-domain" (Aharoni und Goldberg, 2020). Other fields that make use of such data can be cited for information: the healthcare field (Alsentzer u. a., 2019), the scientific field (Beltagy u. a., 2019).

Typically, when working on such fields while using a transformer-based model pre-trained on general English data, there is a drop in performance. This is justified intuitively by the difference in nature between the pre-training corpus textual data and the downstream task in-domain textual data. This phenomenon is called "domain shift".

To mitigate this, a recommended strategy is to incorporate in-domain data in the pre-training step. This can be done in several ways. For instance, we can resume the pre-training of an already pre-trained model on the in-domain data. This is referred to as "continual pre-training" or "mixed pre-training" (Gururangan u. a., 2020).

Another way to operate is to pre-train an untrained model on the data that is similar to the one on which the model will be fine tuned. This strategy is referred to as "pre-training from scratch". According to Gu u. a. (2021), when comparing the different pre-training strategies in a biomedical context which is also characterized by the use

of specialized language, pre-training from scratch obtains better performances than continual or mixed pre-training.
This is why in this article, we chose to favor pre-training from scratch.

2.1.3 Pre-training data

Our pre-training corpus consists of all the textual data available after 2009 because the current reports use the writing style that started from that era, as mentioned in Appendix B. As indicated in Table 2, our pre-training corpus is roughly 2000 times smaller than the one used by RoBERTa.

Model	Custom	RoBERTa
Data	ASRS 2009-2019	Web Crawl
Size	74.5MB	160 GB
Type	Aviation-related language	Standard English

Table 2: Comparison of amount of textual data used for pre-training

2.2 Fine tuning

2.2.1 Fine tuning hyperparameters

For fine tuning, our custom pre-trained model was trained separately on 14 different TC downstream tasks, as detailled in Section 2.2.3. For each fine tuned version of the model, we used the same hyperparameters as Electra-Small for Glue (Wang u. a., 2019) because the size of the two models is comparable. A comparison with the hyperparameters used in the RoBERTa article for Glue is in Table 3.

Hyperparam	Custom	RoBERTa
Learning rate	3e-4	{1e-5,2e-5,3e-5}
Weight Decay	0	0.1
Batch size	32	{16,32}
Max Train Epoch	3	10
Warmup ratio	0.1	0.06

Table 3: Comparison of hyperparameters for fine tuning

2.2.2 Sliding window

The input of our model is limited to 512 tokens, but some of the documents are four times longer than that. To mitigate that, we used the "sliding window" feature of SimpleTransformers (Rajapakse, 2019). When training a model with the sliding window enabled, a document exceeding the limit of input of the classification model will be split into sub-sequences. Each sub-sequence will then be assigned the label from the original sequence and the model will be trained on the full set of sub-sequences.

During evaluation and prediction, the model predicts a label for each window or sub-sequence of an example. The final prediction for a given long document was

originally the mode of the sub-sequences predictions. The authors chose to change the code so that the final prediction was the means of the prediction for each window. This was done to prevent having a default value proposed in the case of a tie.

2.2.3 Fine tuning data

2.2.3.1 The 14 Text Classification datasets

In this Section, we characterize the training and testing datasets used as our text classification (TC) downstream tasks. A TC problem can be defined as the task of choosing for a given text an element from a set of classes or labels. In our case, because we have 14 different TC problems, we must find the correct label for each document from 14 different sets of labels.
Our TC problems are made of one of the 14 subcategories of the "Anomaly" attribute, which belongs to the "Event" entity, following the entity-based corpus structure described in Appendix B. One can see that in most TC problems (see Tables 4 and 8), the proportion of documents per class is very unevenly distributed. Data sparsity is further characterized in the two Tables with the Shanon equitability index. The metric is explained further in Section 4.1.
Some of these TC problems are multi-classification problems, meaning that there are at least 3 classes (see Table 8), others are binary classification problems (see Table 4). For instance, the subcategory Aircraft Equipment Problem can be seen as a multi-classification problem with 3 classes: Critical, Less Severe, and No (meaning "No anomaly of this kind"). The binary classification problems, such as ATC Issue, only has 2 classes: All Types and No.
For each anomaly subcategory, there is a training and testing dataset. In these datasets, the narratives are the inputs, the corresponding subcategory's classes are the expected output.
One of our purposes is to assess the ability of our algorithm to do TC with minimal human-expert intervention. The underlying goal is to make an algorithm that is suitable for use before the incident report has been coded by analysts (this step is seen in the report processing flow from Figure 4). This is why we don't use any analyst-produced metadata or textual data related to the events such as the synopses or callbacks as supplementary input, as it embeds human-expert intervention. This stands in contrast with previous work on the topic of TC on ASRS documents such as Zhang und Mahadevan (2019).
Similar to the pre-training step, we only work on events from after 2009, to avoid having two different styles of writing in our documents. Additionally, we also only use reports produced by reporters from the Flight Crew job category, as done in Darveau u. a. (2020), where the author classifies human factors involved in ASRS narratives. We did that because the Flight Crew job category constitutes 74.2% of the reports and because we work under the assumption that the data would be too heterogeneous if we included other job categories, as shown by the use of different reporting forms.

Name	Composition	Shannon equitability index
ATC Issue	No (86.51%), All Types (13.49%)	0.571
Airspace Violation	No (97.44%), All Types (2.56%)	0.172
Deviation - Speed	No (96.89%), All Types (3.11%)	0.2
Deviation - Track / Heading	No (92.14%), All Types (7.86%)	0.397
No Specific Anomaly Occurred	No (98.77%), All Types (1.23%)	0.096

Table 4: Anomaly subcategories that are binary classification problems

2.2.3.2 Use of ASRS analytical metadata for classification datasets in the literature

When characterizing analyst-produced metadata, one can distinguish between factual and analytical metadata. In Tulechki (2015), the author defined analytical data as metadata that needs "expert inference or reasoning to be produced". In the case of ASRS, the author states that: "Analytical metadata is present both in the form of the Assessment entity and in some of the attributes, such as the Human Factors attributes of the Person entities. ".

In this Section, we contrast and compare our choice of metadata for our TC problems with two other TC problems derived from the analytical part of the ASRS dataset, that were tackled in Darveau u. a. (2020) and Yang (2020).

In Darveau u. a. (2020), the author asks different safety analysts to classify ASRS incidents on the Human Factor (HF) attribute. He also uses an NLP algorithm on that same task. The author finds that both the analysts and the algorithm perform poorly. He finds a high rate of disagreement between analysts. He suggests that the narratives tend to be inconsistently annotated because depending on the human expert's experience, there are differences in how the data is classified. Annotators introducing biases in the datasets (Landis und Koch, 1977), particularly in the case of datasets used by the NLP model for training, have been reported to happen across a wide range of NLP tasks (Geva u. a., 2019; Evert u. a., 2016; Artstein und Poesio, 2008). As a consequence, in Darveau u. a. (2020)'s case, the TC algorithm performance is heavily hindered by the inconsistency of the training data.

In Yang (2020), the author tries to use an NLP algorithm to classify the Primary Problem and Contributing Factors in ASRS incidents, both of which are analytical metadata that belong to the 'Assessment' entity. However, in doing so, the author makes heavy simplifying assumptions: he considers that each incident has always two contributing factors when there can be more or less than two. The author's simplifying assumptions transformed the problem faced by analysts into a simpler one that can be tackled as a TC problem, but there seems to be a trade-off with how practical the model will be for use in the field. Deciding how many contributing factors there is in an incident, seems to be a skill in itself that participates in making the concerned metadata analytical.

Hence, the expert inference needed for coding analytical metadata seems to spur inconsistencies in annotations one one side (Darveau u. a., 2020), and might require to make simplifying assumptions that are seemingly not practical for use in the field on the other side (Yang, 2020).
In contrast, there is no a priori need for simplification of our classification problems. Also, we assume that analysts converge more easily on the identification of the anomalies than on the identification of the HF issue (due to similar training/background). As such, the prediction of the Anomaly attribute of the metadata seems mostly factual. The only noticeable exceptions are for the following anomalies: Equipment Problem and Conflict, where a class comes in two flavors: Critical and Less Severe. We suppose that sorting the anomaly between the two alternatives requires some expert inference, similarly to how the primary problem is chosen among the possible contributing factors for the article (Yang, 2020). However, this seems to be marginal when compared with the expert knowledge needed for predicting human factors or contributing factors.
The reasons above motivated us to work on the "Anomaly" attribute subcategories as opposed to other more complicated analytical metadata.

3 Statistical study of MCC

In this section, we evaluate our model using an empirical score, for which we provide confidence intervals. We also use statistical hypothesis testing to study our model's performance.
Choosing an evaluation metric for a classification procedure is not trivial. Each metrics has its pros and cons (Chicco und Jurman, 2020; Powers, 2020; Yang und Liu, 1999; Sokolova und Lapalme, 2009), and its choice reflects an intent. In our case, we chose the Matthews correlation coefficient (MCC) as opposed to other popular metrics used in our field, such as precision, recall, accuracy, or F1 score.

3.1 General framework

Let $\hat{C} = (\hat{c}_{i,j})_{0 \leq i,j \leq N-1}$ be the empirical confusion square matrix of size K defined by:
Let $\mathcal{T} = (T_n)_{0 \leq n \leq N-1}$ be a corpus of N texts we want to classify into K classes. We have at our disposal a classifier that associate to each text its predicted class k_n where as l_n denotes the true class of the text T_n.
Let $\hat{C} = (\hat{c}_{i,j})_{0 \leq i,j \leq N-1}$ be the empirical confusion square matrix of size K defined by

$$\hat{c}_{i,j} = \sum_{n=0}^{N-1} X_n(i,j) \tag{1}$$

with

$$X_n(i,j) = \mathbb{1}_{k_n=i} \mathbb{1}_{l_n=j}, \tag{2}$$

and let $\hat{P} = (\hat{p_{i,j}})$ be the proportion matrix defined by $\hat{P} = \frac{1}{N}\hat{C}$.

Assumption 1

For all $(i,j) \in \{1,...,K\}^2$, the elements of the sequence $(X_n(i,j))_{0\leq n\leq N-1}$ are i.i.d random variables of the same law as $X(i,j)$ with:

$$\mathbb{P}(X_n(i,j)=1) = 1 - \mathbb{P}(X_n(i,j)=0) = p_{i,j}. \tag{3}$$

Remark 1. *$p_{i,j}$ represents the probability that a text categorized in the i-class belongs to the j-class. By the strong law of large number we obviously have*

$$\hat{p}_{i,j} \xrightarrow[N\to\infty]{a.s.} p_{i,j} \tag{4}$$

3.2 Quantifying the efficiency of the classifying procedure

Definition 1. *The empirical Matthews Correlation Coefficient (MCC) metric is defined by:*

$$\hat{MCC} = \frac{N \times \sum_{k=0}^{K-1} \hat{c}_{k,k} - \sum_{k=0}^{K-1}(\sum_{i=0}^{K-1} \hat{c}_{i,k} \times \sum_{j=0}^{K-1} \hat{c}_{k,j})}{\sqrt{N^2 - \sum_{k=0}^{K-1}(\sum_{i=0}^{K-1} \hat{c}_{i,k})^2} \times \sqrt{N^2 - \sum_{k=0}^{K-1}(\sum_{j=0}^{K-1} \hat{c}_{k,j})^2}}. \tag{5}$$

and the true MCC is defined by

$$MCC_{true} = \frac{\sum_{k=0}^{K-1} p_{k,k} - \sum_{k=0}^{K-1}(\sum_{i=0}^{K-1} p_{i,k} \times \sum_{j=0}^{K-1} p_{k,j})}{\sqrt{1 - \sum_{k=0}^{K-1}(\sum_{i=0}^{K-1} p_{i,k})^2} \times \sqrt{1 - \sum_{k=0}^{K-1}(\sum_{j=0}^{K-1} p_{k,j})^2}}. \tag{6}$$

Remark 2. *Note that by multiplying the numerator and denominator of* (5) *by $\frac{1}{N^2}$:*

$$\hat{MCC} = \frac{\sum_{k=0}^{K-1} \hat{p}_{k,k} - \sum_{k=0}^{K-1}(\sum_{i=0}^{K-1} \hat{p}_{i,k} \times \sum_{j=0}^{K-1} \hat{p}_{k,j})}{\sqrt{1 - \sum_{k=0}^{K-1}(\sum_{i=0}^{K-1} \hat{p}_{i,k})^2} \times \sqrt{1 - \sum_{k=0}^{K-1}(\sum_{j=0}^{K-1} \hat{p}_{k,j})^2}}. \tag{7}$$

Hence, by Equation (4)

$$\hat{MCC} \xrightarrow[N\to\infty]{a.s.} MCC_{true}. \tag{8}$$

The MCC computes the correlation coefficient between the observed categories and the predicted classifications. When the classifier is perfect, we obtain 1. When its output is random, we obtain 0.

MCC is symmetric: in a binary setting, when swapping positive and negative classes, the score will remain the same. That stands in contrast with precision, recall, or F1 score, which ignore performance in handling negative examples. Besides, its score is not biased towards classes with a larger size, as the accuracy metric. To obtain a good MCC score, the classifier must be good on every class that constitutes a classification problem, regardless of the dataset class imbalances. We aimed to evaluate our model's overall performance on classification problems, as opposed to its performance on a single class. This is our strongest argument for using MCC.

In practice, we only have access to the empirical value $\hat{MCC}$, it is then natural to quantify how far $\hat{MCC}$ is from MCC_{true}. One classical way to answer this question is to provide a confidence interval for MCC_{true}. Moreover, if one wants to compare two different classification procedures, one can compute for each procedure the associated $\hat{MCC}_1$ and $\hat{MCC}_2$, and investigate whether there is statistical evidence that:

$$\hat{MCC}_1 \leq \hat{MCC}_2 (resp. \geq) \implies MCC_{true1} \leq MCC_{true2} (resp. \geq).$$

This can be done thanks to a confidence interval for $\hat{MCC}_1 - \hat{MCC}_2$ which is classically derived from a joint Central Limit Theorem for $(\hat{MCC}_1, \hat{MCC}_2)$. To the best of the authors' knowledge at the time of writing, we have not seen articles using confidence intervals for MCC scores, to provide a sense of what the precision of the results is.

For each classification problem, we also investigated whether there is statistical evidence that our MCC differs strongly depending on the usage of the sliding window feature (which depends on the document's length), using hypothesis testing.

3.2.1 Vectorial central limit theorem for $\hat{MCC}$ and application to confidence interval

Let P_n be the square matrix of size K, where the variable at the column j and line i is $X_n(i,j)$. We obtain that

$$\hat{P} = \frac{1}{N}\sum_{n=0}^{N-1} P_n. \tag{9}$$

Let $\mathcal{M}_K$ be the space of all the square real matrices of size K, and g be the application defined by:

$$\begin{array}{rccc} g & : & \mathcal{M}_K & \to & \mathbb{R} \\ & & (x_{i,j}) & \mapsto & g((x_{i,j})), \end{array}$$

where

$$g((x_{i,j})) = \frac{\sum_{k=0}^{K-1} x_{k,k} - \sum_{k=0}^{K-1}(\sum_{i=0}^{K-1} x_{i,k} \times \sum_{j=0}^{K-1} x_{k,j})}{\sqrt{1 - \sum_{k=0}^{K-1}(\sum_{i=0}^{K-1} x_{i,k})^2} \times \sqrt{1 - \sum_{k=0}^{K-1}(\sum_{j=0}^{K-1} x_{k,j})^2}}. \tag{10}$$

Theorem 3.1. *If Assumption 1 holds, then*

1. $$\sqrt{N}((\hat{MCC} - MCC_{true}) \xrightarrow[N\to\infty]{\mathcal{L}} \mathcal{N}_1(0\,, \sigma_1), \tag{11}$$

   where $\sigma_1 = \sqrt{Dg^t \Sigma Dg}$, *and* Σ *is the covariance matrix of size* K^2

   $$\Sigma = \begin{bmatrix} Cov(X(0,0), X(0,0)) & \dots & Cov(X(0,0), X(K-1,K-1)) \\ \dots & \dots & \dots \\ Cov(X(K-1,K-1), X(0,0)) & \dots & Cov(X(K-1,K-1), X(K-1,K-1)) \end{bmatrix},$$

   and $Dg = (\frac{\partial g}{\partial x_{i,j}})_{(i,j)\in\{0,K-1\}^2}$ *is the gradient at the coordinates* $\hat{P}$ *of the application g.*

2. *Let* $0 \leq \alpha \leq 1$ *and* $\hat{\sigma_1}$ *a consistent estimator of* σ_1, *then*

   $$\lim_{N\to\infty} \mathbb{P}\left(\hat{MCC} - \frac{q_\alpha \times \hat{\sigma}_1}{\sqrt{N}} \leq MCC_{true} \leq \hat{MCC} + \frac{q_\alpha \times \hat{\sigma}_1}{\sqrt{N}}\right) = 1 - \alpha \tag{12}$$

   where q_α *is such that* $\mathbb{P}\left(-q_\alpha \leq \mathcal{N}(0,1) \leq q_\alpha\right) = 1 - \alpha$.

Remark 3. *In particular, the confidence interval with a 95% confidence level is given by*

$$]\hat{MCC} - \frac{1,96 \times \hat{\sigma}_1}{\sqrt{N}}, \hat{MCC} + \frac{1,96 \times \hat{\sigma}_1}{\sqrt{N}}[. \tag{13}$$

Remark 4. *Proof of theorem and closed formula for* Dg *and* $\hat{\Sigma}$ *are given in Appendix H.*

3.2.2 Application to statistical test

In this Section, we aim to decide whether there is statistical evidence that the model's performance represented by MCC, is affected by characteristics of our input such as document length. To do so, we test $H_0 : H_0 : MCC_{true1} = MCC_{true2}$ against $H_1 : H_1 : MCC_{true1} \neq MCC_{true2}$, where MCC_{true1} is the true MCC value for a part of the corpus, and MCC_{true2} is the MCC value for the other distinct part of the corpus. The distinction is based on the characteristic that is being evaluated. For a given characteristic, let δ_n be the variable associated to the characteristic, such that for each text T_n, $\delta_n = 1$ if and only if the text shares that characteristic. This is used to formalize MCC values calculated on a subset of the corpus based on a characteristic. Let

$$X1_n(i,j) = X_n(i,j)(1 - \mathbb{1}_{\delta_n=1}) \tag{14}$$
$$X2_n(i,j) = X_n(i,j)\mathbb{1}_{\delta_n=1} \tag{15}$$

Let $Q1_n$ (resp. $Q2_n$)be the square matrix of size K, where the variable at the column j and line i is $X1_n(i,j)$ (resp. $X2_n(i,j)$).

$$Q1_n = (1 - \mathbb{1}_{\delta_n=1}) \times P_n, \tag{16}$$
$$Q2_n = \mathbb{1}_{\delta_n=1} \times P_n. \tag{17}$$

Let U_n be the vector of such that U_n is the juxtaposition of $Q1_n$, $Q2_n$ and $\mathbb{1}_{\delta_n=1}$. Let $\overline{U}_N$ be defined by:

$$\overline{U}_N = \frac{1}{N}\sum_{n=0}^{N-1} U_n. \tag{18}$$

We define the functions J and f by

$$\begin{array}{rccc} J & : & M_K \times M_K \times \mathbb{R} & \longrightarrow & \mathbb{R}^2 \\ & & (x,y,z) & \longmapsto & (g(\frac{x}{1-z}), g(\frac{y}{z})). \\ f & : & \mathbb{R}^2 & \longrightarrow & \mathbb{R} \\ & & (x,y) & \longmapsto & x - y. \end{array} \tag{19}$$

Theorem 3.2. *Under H_0,*

$$\sqrt{N}(\hat{MCC}_1 - \hat{MCC}_2) \xrightarrow[N\to\infty]{\mathcal{L}} \mathcal{N}(0\,, \sigma_2^2). \tag{20}$$

where

$$\sigma_2 = \sqrt{Df^t \Sigma_b Df}, \tag{21}$$

Df is the gradient of the application f, and $\Sigma_b = DJ^t \Sigma_a DJ$ with DJ the gradient of the application J, and $\Sigma_a = Cov(U_n)$ is a covariance matrix of size $2K^2 + 1$.

Remark 5. *Hence, we will reject H_0 as soon as $|\hat{MCC}_1 - \hat{MCC}_2| > q_\alpha$. The threshold q_α is qualified thanks to Theorem 3.2 and the slutsky Lemma. For example the threshold for a test of level $\alpha = 5\%$ is given by $\frac{1.96\sigma_2}{\sqrt{N}}$.*

Remark 6. *Note that the framewrok developped in Section 3 can be applied on any of the classical metrics such as precision, recall, or f1 score.*

Remark 7. *Proof of theorem and closed formula for DJ and Σ_a are in Appendix H.*

4 Results

4.1 Model performance with confidence interval

In Table 5, we give the measures of MCC with a 95% confidence interval for our algorithm, as obtained on the testing dataset for the different classification problems.

anomaly	MCC with confidence interval
ATC Issue	0.63 +/- 0.051
Aircraft Equipment Problem	0.577 +/- 0.041
Airspace Violation	0.552 +/- 0.14
Conflict	0.704 +/- 0.044
Deviation - Altitude	0.418 +/- 0.057
Deviation - Procedural	0.39 +/- 0.04
Deviation - Speed	0.609 +/- 0.118
Deviation - Track / Heading	0.681 +/- 0.066
Flight Deck / Cabin / Aircraft Event	0.585 +/- 0.059
Ground Event / Encounter	0.586 +/- 0.062
Ground Excursion	0.809 +/- 0.08
Ground Incursion	0.505 +/- 0.111
Inflight Event / Encounter	0.513 +/- 0.041
No Specific Anomaly Occurred	0.0 +/- nan
size sample	1154

Table 5: MCC of the custom model on the different testing datasets

Because the MCC score can be defined as a special case of Pearson Correlation Coefficient (Powers, 2020), it can be interpreted in the same way.

We notice that for the anomaly "Ground excursion", there is a very strong correlation between the predicted classes and the labels (> 0.7), even when taking into account the confidence interval.

For all the other anomalies except "Deviation - Procedural", we notice a strong correlation between the predicted classes and the labels (> 0.4). This remains true even when taking into account the confidence interval, except for the anomaly attribute subcategories "Ground Incursion" and "Deviation - Altitude", for which the confidence interval lower bound is lower than 0.4.

In the specific case of the anomaly "No Specific Anomaly Occurred", the model's predictions are no better than random, with an MCC score of 0. Our intuition is that this is the result of an extremely unbalanced dataset, as can be seen in the confusion matrix Figure 1 (all the confusion matrices are in fig 5 in Appendix F).

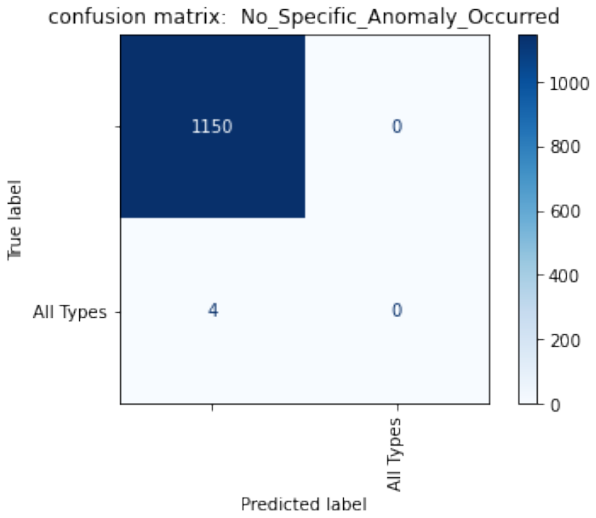


Figure 1: Confusion matrix for "No Specific Anomaly Occurred"

Overall, the model performs relatively well.
We can observe that the confidence interval for our MCC scores varies a lot. There seems to be a negative correlation between the data sparsity of the different classification problems and the confidence interval values. To visualize this, we use the "Shannon equitability index" (Fath, 2018):

$$E_H = \frac{H}{log(K)}, \tag{2}$$

with:

$$H = -\sum_{i=1}^{K} \frac{c_i}{n} log(\frac{c_i}{n}), \tag{3}$$

and where K is the number of classes, and c_i is the number of elements in class i. The result is a number between 0 and 1. The higher the score, the more balanced is the dataset. The scores obtained for each of the datasets are given in Tables 4 and 8.

We observe a non-linear monotonic negative correlation between the Shannon equitability index and ci, which is the range of the confidence interval divided by 2, in the scatter-plot of Figure 2:

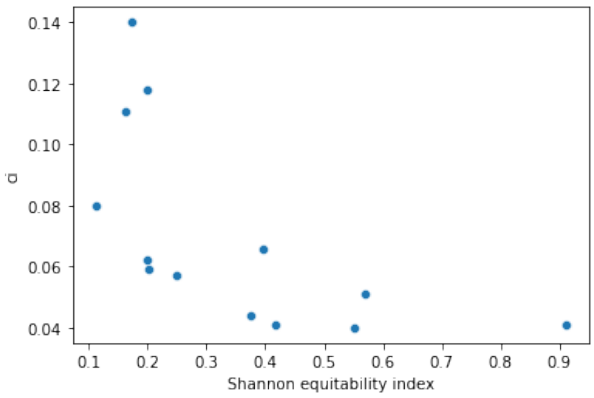


Figure 2: ci against Shanon equitability index of the classification problems

We obtain a strong Spearman correlation score of -0.82. This result has to be nuanced by the small size of our sample, however, it would seem that the more unbalanced a dataset is, the larger the confidence interval will be for the MCC score.

4.2 Impact of the sliding window

In Table 6, we give for each anomaly the MCC score for the sub-corpora where the documents are of size windows=1 ($\hat{MCC}_1$), and the MCC score for the sub-corpora where the documents are of size windows>1 ($\hat{MCC}_2$). In the latter case, the documents are long enough to prompt the model to use the mechanism of the sliding window described in Section 2.2.2, to make a prediction. We also give in the Table the associated *threshold* and result for the following statistical test:

$$H_0 : \hat{MCC}_1 = \hat{MCC}_2$$
$$H_1 : \hat{MCC}_1 \neq \hat{MCC}_2$$

When $threshold < |\hat{MCC}_1 - \hat{MCC}_2|$, we decide H1 with a significance level of 0.05 (meaning that we have a 5% risk of concluding that a difference exists when there is no actual difference). If this not the case, we cannot conclude if either hypothesis is true. We notice that in most cases, we do not find strong statistical evidence that our model performs differently depending on the usage of the sliding windows mechanism or the lack thereof.

anomaly	windows=1	windows>1	threshold	decision
ATC Issue	0.618	0.685	0.126	-
Aircraft Equipment Problem	0.62	0.519	0.076	H1
Airspace Violation	0.522	0.701	0.293	-
Conflict	0.726	0.63	0.117	-
Deviation - Altitude	0.438	0.348	0.157	-
Deviation - Procedural	0.402	0.307	0.104	-
Deviation - Speed	0.607	0.651	0.25	-
Deviation - Track / Heading	0.703	0.619	0.164	-
Flight Deck / Cabin / Aircraft Event	0.599	0.614	0.156	-
Ground Event / Encounter	0.628	0.376	0.177	H1
Ground Excursion	0.827	0.668	0.313	-
Ground Incursion	0.493	0.581	0.274	-
Inflight Event / Encounter	0.543	0.408	0.101	H1
No Specific Anomaly Occurred	0.0	0.0	-	-
size sample	926	228	-	-

Table 6: Results of hypothesis testing

Similar to the case of the confidence interval, we observe in Figure 3 a negative correlation between the different classification problem testing datasets' imbalances and the threshold values:

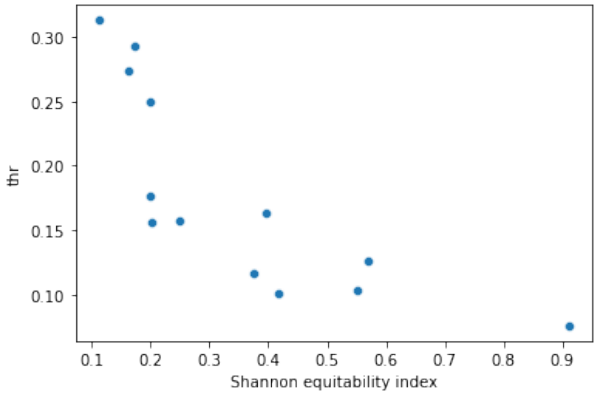


Figure 3: Threshold values against Shanon equitability index of the classification problems

We obtain a strong Spearman correlation score of -0.91. This result has to be nuanced by the small size of our sample however, it would seem that the more unbalanced a dataset is, the bigger the threshold will be for the hypothesis testing.

5 Discussion

5.1 Hypothesis testing

Hypothesis testing could be further used to evaluate the impact of any distinctive feature of an incident report on the classification model performance. This can be especially useful for documents that have metadata.
As an example, one could investigate if the model performs significantly differently on incidents depending on the value of reporter-based metadata, such as the flight phase. This provides a simple tool to add understanding on what aspect of an incident impacts the model performance.

5.2 Extrinsic vs intrinsic evaluation

When evaluating an NLP system or model, there are two distinct approaches. An intrinsic evaluation focuses on evaluating the model within the context of the formal task it is given. This stands in contrast with extrinsic evaluation, which aims at evaluating the model within the global context of its usage environment (Jones und Galliers, 1995; Tanguy u. a., 2016; Ittoo u. a., 2016).
In the case of TC adapted from semi-structured aviation safety database such as ASRS, Tulechki (2015) says that there is a strong overlap between extrinsic and intrinsic considerations when evaluating the models used to tackle the task. Intuitively, this is because an intrinsically good TC model has better chances to help analysts do their work. For instance, in the hypothetical case of a model that scores MCC scores of 1 on each TC problem, the analyst is no longer needed to code documents and can devote his time to other tasks. In the case of our work, however, the way we evaluated our model lacks a few elements to be considered fully extrinsic.
First, as stated in Section 3.1, the choice of metric reflects an intent on the side of the assessor. In real-world case scenarios, from the detection of diabetes in patients to the classifying of emails as spam, the cost associated with a wrong prediction often differs depending on the class. For instance, wrongly predicting that a patient has diabetes is less consequential than wrongly predicting that a patient does not have diabetes. This is where metrics like precision or recall are useful (Yang und Liu, 1999; Sokolova und Lapalme, 2009), because they give information on our model's performance on a class basis, as opposed to the performance on every class at the same time.
For a given class, precision is the ratio between the number of times the class was correctly predicted and the number of times the class was predicted. In contrast, recall is the ratio between the number of times the class was correctly predicted and the number of times the class should have been predicted if the model was perfect. For a given class, a high recall means that our model will correctly classify most of the documents that have the corresponding label, whereas a high precision means that our model has a high chance to be accurate when it predicts the concerned label.
There is usually a trade-off between the two, and which one to use to assess our model depends on the problem. For instance, a high recall is more important than a high precision in the case where experts prefer to run the risk of a model predicting wrongly there was an anomaly, rather than to run the risk of missing the anomaly altogether.
Once a metric or a set of metrics has been chosen to characterize the model's performance, threshold values can help determine a level of trust in our prediction during inference, or field use. For instance, in Tulechki (2015), the author requires a threshold

of 95% of precision for accepting a prediction of a model. It means that for a given class, if the model has a precision that is estimated to be above 95% during the evaluation phase, the model is trusted when it predicts the class. However, if the model predicts a class for which the precision is estimated below 95% during the evaluation phase, then human intervention is needed and the model can be used as a suggestion system instead or simply ignored.
As an example, we provide in Table 7 the results for the discussed metrics on the label "no anomaly of this kind" for each of the text classification problems. Hypothetically, if we used the same standards as Tulechki (2015) of having a threshold of 95% of precision, then we would trust our model when it predicts that there is "no anomaly of this kind" on every TC problem except for "Deviation - Procedural" and "Aircraft Equipment Problem", because the related precision scores fall under the threshold. In this last case, a human safety expert would be solicited to categorize the text by hand. Arguably, precision might not be the best indicator, because we do not account for the risk of missing an event. Hypothetically, if we applied the same threshold of 95% to recall, there would be 50% fewer cases where our model could be trusted with predicting the "no anomaly of this kind" label: 6 instead of 12 (Airspace Violation, Deviation - Speed, Ground Event / Encounter, Ground Excursion, Ground Incursion, No Specific Anomaly Occurred). This shows how the choice of metric can impact the validity of our model.

anomaly	precision	recall	f1
ATC Issue	0.976	0.869	0.919
Aircraft Equipment Problem	0.924	0.899	0.911
Airspace Violation	0.993	0.98	0.986
Conflict	0.987	0.929	0.957
Deviation - Altitude	0.988	0.898	0.94
Deviation - Procedural	0.713	0.703	0.708
Deviation - Speed	0.991	0.978	0.984
Deviation - Track / Heading	0.988	0.942	0.965
Flight Deck / Cabin / Aircraft Event	0.981	0.902	0.94
Ground Event / Encounter	0.958	0.962	0.96
Ground Excursion	0.995	0.987	0.991
Ground Incursion	0.994	0.977	0.985
Inflight Event / Encounter	0.938	0.757	0.838
No Specific Anomaly Occurred	0.997	1.0	0.998

Table 7: Precision, recall and F1 score for category "no anomaly of this kind"

We intentionally chose to use the MCC metric, because we did not have access to ASRS safety analysts to conjointly determine a metric or set of metrics of evaluation that were adapted to the "business" problem at hand. In this case, we assumed the next best thing to do was to evaluate our model overall performance, for which precision and recall are bad indicators, similarly to the F1 score which is simply the harmonic mean of precision and recall, for the reasons that were explained in Section 3.1.
To conclude, our intrinsic evaluation provides a good indication of our model overall

performance, which is made even more precise by the use of confidence intervals, but does not provide a good extrinsic indication of our model's usefulness or reliability (8016712, 2017) in the field, as we lack a clear idea of what are the requirements for such models.
We suggest for future work, that a choice of metric and a corresponding per-class basis value for performance threshold, should be discussed directly with the ASRS analysts that code the reports, as well as their perceptions of the results and usefulness of the technology.

6 Conclusion

In this article, we did not challenge the inherent value of the written event reports, as well as the associated metadata, but instead, we focused on using TC techniques to support their categorization. Specifically, we have used a custom language model, trained from scratch on domain-specific data, to classify event reports on 14 classification problems. A summary of the NLP pipeline can be found in Appendix I.
We have provided arguments for choosing the MCC score as our evaluation metric for our model performance, as well as shown how to get a confidence interval. We have proposed a way to use hypothesis testing to evaluate if our model performed differently on two distinct sub-corpora. In our case, the distinction was based on the length of the documents, which prompted the use of the mechanism of the sliding window to produce a prediction for long documents.
Finally, we have discussed what it would require for the model to be potentially used in the industry. Our model is freely available for use on Huggingface (Wolf u. a., 2020), under the name ASRS-CMFS (Aviation Safety Reporting System - Compact Model trained From Scratch). We leave for future work the comparison of our model's performance with other models.

Acknowledgement

The authors would like to thank Corinne Bieder for advising on the article.

Funding

This research was supported by the ENAC – AIRBUS Safety Management chair.

References

[8016712 2017] ISO/IEC/IEEE International Standard - Systems and software engineering–Vocabulary. In: *ISO/IEC/IEEE 24765:2017(E)* (2017), S. 1–541

[Aharoni und Goldberg 2020] Aharoni, Roee ; Goldberg, Yoav: Unsupervised Domain Clusters in Pretrained Language Models. In: *CoRR* abs/2004.02105 (2020). – URL `https://arxiv.org/abs/2004.02105`

[Alsentzer u. a. 2019] Alsentzer, Emily ; Murphy, John ; Boag, William ; Weng, Wei-Hung ; Jindi, Di ; Naumann, Tristan ; McDermott, Matthew: Publicly Available Clinical BERT Embeddings. In: *Proceedings of the 2nd Clinical Natural Language Processing Workshop*. Minneapolis, Minnesota, USA : Association for Computational Linguistics, Juni 2019, S. 72–78. – URL https://aclanthology.org/W19-1909

[Artstein und Poesio 2008] Artstein, Ron ; Poesio, Massimo: Inter-Coder Agreement for Computational Linguistics. In: *Computational Linguistics* 34 (2008), 12, Nr. 4, S. 555–596. – URL https://doi.org/10.1162/coli.07-034-R2. – ISSN 0891-2017

[ASRS 2019] ASRS: *ASRS Program Briefing*. 2019. – URL https://asrs.arc.nasa.gov/docs/ASRS$_{P}rogramBrief ing.pdf$. – $-Zugriffsdatum$: 2019 − 07 − 01

[ASRS 2021] ASRS: *ASRS reporting form*. 2021. – URL https://asrs.arc.nasa.gov/report/electronic.html. – Zugriffsdatum: 2021-02-23

[Beltagy u. a. 2019] Beltagy, Iz ; Lo, Kyle ; Cohan, Arman: SciBERT: A Pretrained Language Model for Scientific Text. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China : Association for Computational Linguistics, November 2019, S. 3615–3620. – URL https://aclanthology.org/D19-1371

[Chicco und Jurman 2020] Chicco, Davide ; Jurman, Giuseppe: The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. In: *BMC Genomics* 21 (2020), 01

[Clark u. a. 2020] Clark, Kevin ; Luong, Minh-Thang ; Le, Quoc V. ; Manning, Christopher D.: *ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators*. 2020

[Darveau u. a. 2020] Darveau, Katherine ; Hannon, Daniel ; Foster, Chad: A Comparison of Rule-Based and Machine Learning Models for Classification of Human Factors Aviation Safety Event Reports. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 64 (2020), 12, S. 129–133

[Devlin u. a. 2019] Devlin, Jacob ; Chang, Ming-Wei ; Lee, Kenton ; Toutanova, Kristina: *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019

[Evert u. a. 2016] Evert, Stefan ; Greiner, Paul ; Baigger, João F. ; Lang, Bastian: A distributional approach to open questions in market research. In: *Computers in Industry* 78 (2016), S. 16–28. – URL https://www.sciencedirect.com/science/article/pii/S016636151530049X. – Natural Language Processing and Text Analytics in Industry. – ISSN 0166-3615

[Fath 2018] Fath, Brian D.: *Encyclopedia of ecology*. Elsevier, 2018

[Geva u. a. 2019] Geva, Mor ; Goldberg, Yoav ; Berant, Jonathan: Are We Modeling the Task or the Annotator? An Investigation of Annotator Bias in Natural Language Understanding Datasets. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China : Association for Computational Linguistics, November 2019, S. 1161–1166. – URL https://aclanthology.org/D19-1107

[Gu u. a. 2021] Gu, Yu ; Tinn, Robert ; Cheng, Hao ; Lucas, Michael ; Usuyama, Naoto ; Liu, Xiaodong ; Naumann, Tristan ; Gao, Jianfeng ; Poon, Hoifung: *Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing*. 2021

[Gururangan u. a. 2020] Gururangan, Suchin ; Marasović, Ana ; Swayamdipta, Swabha ; Lo, Kyle ; Beltagy, Iz ; Downey, Doug ; Smith, Noah A.: Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online : Association for Computational Linguistics, Juli 2020, S. 8342–8360. – URL https://aclanthology.org/2020.acl-main.740

[Howard und Ruder 2018] Howard, Jeremy ; Ruder, Sebastian: *Universal Language Model Fine-tuning for Text Classification*. 2018

[Ittoo u. a. 2016] Ittoo, Ashwin ; Nguyen, Le M. ; van den Bosch, Antal: Text analytics in industry: Challenges, desiderata and trends. In: *Computers in Industry* 78 (2016), S. 96–107. – URL https://www.sciencedirect.com/science/article/pii/S0166361515300646. – Natural Language Processing and Text Analytics in Industry. – ISSN 0166-3615

[Jones und Galliers 1995] Jones, Karen S. ; Galliers, Julia R.: *Evaluating natural language processing systems: An analysis and review*. Bd. 1083. Springer Science & Business Media, 1995

[Kaplan u. a. 2020] Kaplan, Jared ; McCandlish, Sam ; Henighan, Tom ; Brown, Tom B. ; Chess, Benjamin ; Child, Rewon ; Gray, Scott ; Radford, Alec ; Wu, Jeffrey ; Amodei, Dario: *Scaling Laws for Neural Language Models*. 2020

[Kierszbaum 2021] Kierszbaum, Samuel: *information on pre-training of ASRS compact model*. 2021. – URL https://wandb.ai/sam$_e$nac/$ASRS\%20-\%20RoBERTa-SMALL?workspace=user-sam_enac.--Zugriffsdatum$: $2021-07-13$

[Landis und Koch 1977] Landis, J R. ; Koch, Gary G.: The measurement of observer agreement for categorical data. In: *biometrics* (1977), S. 159–174

[Liu u. a. 2019] Liu, Yinhan ; Ott, Myle ; Goyal, Naman ; Du, Jingfei ; Joshi, Mandar ; Chen, Danqi ; Levy, Omer ; Lewis, Mike ; Zettlemoyer, Luke ; Stoyanov, Veselin: *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019

[Micheli u. a. 2020] Micheli, Vincent ; d'Hoffschmidt, Martin ; Fleuret, François: On the importance of pre-training data volume for compact language models. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural*

Language Processing (EMNLP), Association for Computational Linguistics, 2020. – URL https://doi.org/10.18653/v1/2020.emnlp-main.632

[Paszke u. a. 2019] Paszke, Adam ; Gross, Sam ; Massa, Francisco ; Lerer, Adam ; Bradbury, James ; Chanan, Gregory ; Killeen, Trevor ; Lin, Zeming ; Gimelshein, Natalia ; Antiga, Luca ; Desmaison, Alban ; Kopf, Andreas ; Yang, Edward ; DeVito, Zachary ; Raison, Martin ; Tejani, Alykhan ; Chilamkurthy, Sasank ; Steiner, Benoit ; Fang, Lu ; Bai, Junjie ; Chintala, Soumith: PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Wallach, H. (Hrsg.) ; Larochelle, H. (Hrsg.) ; Beygelzimer, A. (Hrsg.) ; Alché-Buc, F. d'(Hrsg.) ; Fox, E. (Hrsg.) ; Garnett, R. (Hrsg.): *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, S. 8024–8035. – URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-libra

[Pellegrino 2019] Pellegrino, Francesca: *The Just Culture Principles in Aviation Law*. Springer International Publishing, 2019. – URL https://doi.org/10.1007/978-3-030-23178-1

[Peters u. a. 2018] Peters, Matthew E. ; Neumann, Mark ; Iyyer, Mohit ; Gardner, Matt ; Clark, Christopher ; Lee, Kenton ; Zettlemoyer, Luke: *Deep contextualized word representations*. 2018

[Powers 2020] Powers, David M. W.: *Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation*. 2020

[Raffel u. a. 2020] Raffel, Colin ; Shazeer, Noam ; Roberts, Adam ; Lee, Katherine ; Narang, Sharan ; Matena, Michael ; Zhou, Yanqi ; Li, Wei ; Liu, Peter J.: *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2020

[Rajapakse 2019] Rajapakse, T. C.: *Simple Transformers*. https://github.com/ThilinaRajapakse/simpletransformers. 2019

[Robinson 2019] Robinson, S.D.: Temporal topic modeling applied to aviation safety reports: A subject matter expert review. In: *Safety Science* 116 (2019), S. 275–286. – URL https://www.sciencedirect.com/science/article/pii/S0925753518308348. – ISSN 0925-7535

[Sokolova und Lapalme 2009] Sokolova, Marina ; Lapalme, Guy: A systematic analysis of performance measures for classification tasks. In: *Information processing & management* 45 (2009), Nr. 4, S. 427–437

[Tanguy u. a. 2016] Tanguy, Ludovic ; Tulechki, Nikola ; Urieli, Assaf ; Hermann, Eric ; Raynal, Céline: Natural language processing for aviation safety reports: From classification to interactive analysis. In: *Computers in Industry* 78 (2016), S. 80–95. – URL https://www.sciencedirect.com/science/article/pii/S0166361515300464. – Natural Language Processing and Text Analytics in Industry. – ISSN 0166-3615

[Tulechki 2015] Tulechki, Nikola: *Natural language processing of incident and accident reports : application to risk management in civil aviation*, Université Toulouse le Mirail - Toulouse II, phdthesis, September 2015. – URL https://tel.archives-ouvertes.fr/tel-01230079. – Zugriffsdatum: 2020-04-21

[Turc u. a. 2020] Turc, Iulia ; Chang, Ming-Wei ; Lee, Kenton ; Toutanova, Kristina: *Well-Read Students Learn Better: On the Importance of Pre-training Compact Models*. 2020. – URL https://openreview.net/forum?id=BJg7x1HFvB

[Vaswani u. a. 2017] Vaswani, Ashish ; Shazeer, Noam ; Parmar, Niki ; Uszkoreit, Jakob ; Jones, Llion ; Gomez, Aidan N. ; Kaiser, Lukasz ; Polosukhin, Illia: *Attention Is All You Need*. 2017

[Wang u. a. 2020] Wang, Alex ; Pruksachatkun, Yada ; Nangia, Nikita ; Singh, Amanpreet ; Michael, Julian ; Hill, Felix ; Levy, Omer ; Bowman, Samuel R.: *SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems*. 2020

[Wang u. a. 2019] Wang, Alex ; Singh, Amanpreet ; Michael, Julian ; Hill, Felix ; Levy, Omer ; Bowman, Samuel R.: *GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding*. 2019

[Wolf u. a. 2020] Wolf, Thomas ; Debut, Lysandre ; Sanh, Victor ; Chaumond, Julien ; Delangue, Clement ; Moi, Anthony ; Cistac, Pierric ; Rault, Tim ; Louf, Rémi ; Funtowicz, Morgan ; Davison, Joe ; Shleifer, Sam ; Platen, Patrick von ; Ma, Clara ; Jernite, Yacine ; Plu, Julien ; Xu, Canwen ; Scao, Teven L. ; Gugger, Sylvain ; Drame, Mariama ; Lhoest, Quentin ; Rush, Alexander M.: Transformers: State-of-the-Art Natural Language Processing. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online : Association for Computational Linguistics, Oktober 2020, S. 38–45. – URL https://www.aclweb.org/anthology/2020.emnlp-demos.6

[Yang 2020] Yang, Qiwei: *Identify Incident Factors to Support Aviation Safety Decision-Making: Proposing a Deep Learning Approach*, The University of Texas at San Antonio, Dissertation, 2020

[Yang und Liu 1999] Yang, Yiming ; Liu, Xin: A re-examination of text categorization methods. In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999, S. 42–49

[Zhang und Mahadevan 2019] Zhang, Xiaoge ; Mahadevan, Sankaran: Ensemble machine learning models for aviation incident risk prediction. In: *Decision Support Systems* 116 (2019), S. 48–63. – URL https://www.sciencedirect.com/science/article/pii/S0167923618301660. – ISSN 0167-9236

[Zhang u. a. 2021] Zhang, Xiaoge ; Srinivasan, Prabhakar ; Mahadevan, Sankaran: Sequential deep learning from NTSB reports for aviation safety prognosis. In: *Safety Science* 142 (2021), S. 105390. – URL https://www.sciencedirect.com/science/article/pii/S0925753521002344. – ISSN 0925-7535

APPENDIX : Hardware

Authors used a servor hosting 8 GeForce RTX 2080 Ti GPUs for computing power.

APPENDIX B: Extensive description of the ASRS dataset

Since its creation, ASRS has received 1,625,738 incident reports up to July 2019 (ASRS, 2019). In 2019, the average number of reports received per week was 2 248, however, not all the received reports are available in the public database.
The full report processing flow is described in Figure 4, and more information can be found in ASRS (2019).
ASRS is a semi-structured dataset with both textual data as well as metadata. Information can be broken down into the following: narratives, synopses, and callbacks for the textual data, reporter-generated, and analyst-generated for the metadata.
The narratives are descriptions of an incident occurrence by a reporter. The synopses are a summary of that description and are written by analysts. Callbacks are supplementary information given by reporters on the occurrence, upon being called back by analysts seeking further details.
The reporter-produced metadata consists of structured information on the incident context (for instance, information on the weather or the pilot experience). It is provided upon completion of the reporting form.
There are four different types of forms, based on the job of the reporter: the Air Traffic Control (ATC) reporting form, the cabin reporting form, the maintenance form, and the general form (ASRS, 2021).
The analysts also produce metadata after receiving the reports. This step referred to as "coding", provides a structured assessment of the occurrences from a safety point of view. This analyst-produced metadata is used to query the dataset, as well as to produce statistics on aviation incidents (Tulechki, 2015).

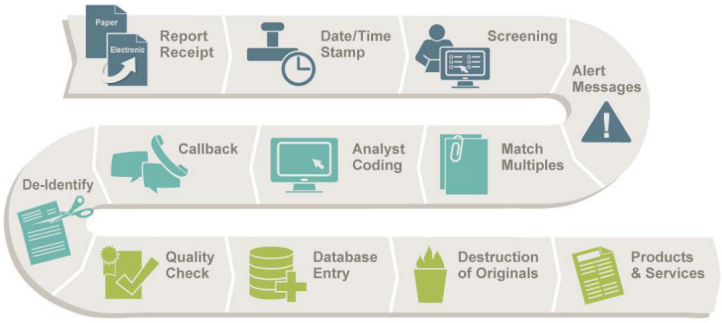


Figure 4: Report Processing Flow, extracted from (ASRS, 2019), p16

The metadata follows a taxonomy, which is built around an entity-based structure (Tulechki, 2015). An example of a report with its metadata is available in Appendix D. At the top level of the taxonomy are entities (Time, Place, Environment, Aircraft, Component, Person, Events, and Assessments). At the lowest branch of the taxonomy are metadata and their values. In between those two, there are entities' attributes and in some cases, attributes' subcategories.

For instance, the insightful"Events" entity has an attribute called "Anomaly", this anomaly has 14 sub-categories, and one of the subcategories "ATC anomaly", can take two possible values: "No", or "All types" .
These 14 subcategories will constitute our classification problems on which we will fine tune our model, as described further in Section 2.2.3.
The textual data-style from 1987 to 2008 included is vastly different from the style used after. Documents from this era are characterized by upper-case letters, fragmented sentences (missing words), and heavy use of abbreviations. Documents after this era have both upper and lower-case letters, sentences are not fragmented, and the use of abbreviations is standardized. A sample of both kinds is provided in appendices D and E.
For information, the dataset received has data ranging from 1987 to 2019. The size of the data is 287 MB, with a total of 385 492 documents and 50 204 970 space-delimited words. In our experiments, we only used data from after 2009. The resulting dataset is composed of roughly 40 000 documents.

APPENDIX C: Introduction to transformer-based algorithms

In 2017, the article "Attention is all you need" describes a neural network architecture that leverages an attention mechanism on translation tasks, the transformer (Vaswani u. a., 2017). Soon, many more architectures adapted from the transformer emerge, each time obtaining better results on Natural Language Understanding (NLU) benchmarks (Wang u. a., 2019), to the point that they even beat the human baseline (Wang u. a., 2020).
Among them is BERT (Devlin u. a., 2019), which obtains state-of-the-art results in many NLU tasks when published. After BERT, another model, RoBERTa (Liu u. a., 2019) emerges, which re-uses the architecture of BERT but changes how the model is trained, once again obtaining state-of-the-art results when published. In Section 2.1.1, we give details on how we implemented our transformer-based model based on the RoBERTa model.
Transformer-based models all have a common characteristic, aside from the use of an attention mechanism. Their training typically involves two steps: pre-training and fine tuning:

- pre-training: Pre-training is a step where the algorithm is trained on a massive amount of unlabelled textual data (Peters u. a., 2018). This task is unsupervised, meaning that there is no need for human annotation. It is typically done by initially corrupting text, for instance by randomly removing words. Then, the algorithm is tasked with restoring the data to its original state. The underlying idea is that the algorithm will learn to model the language through this task. Algorithms that use this kind of training are referred to as language models. In Section 2.1.2, we show that there are different pre-training strategies in cases where the language is domain-specific, such as in aviation, and justify our choice of pre-training strategy for our model.
- fine tuning: Once we have a fully pre-trained model, we can re-train it on other tasks such as document classification (Howard und Ruder, 2018). These tasks are referred to as downstream tasks and are typically supervised and use dramatically fewer data. This retraining procedure is called fine tuning.

The underlying idea behind using these two steps is that the algorithm will leverage what he learned during the pre-training step to obtain better performances on the downstream task during the fine tuning step. This idea is referred to as transfer learning.
A drawback of such methods is that the models used are heavy in size and need a lot of data and computation resources during pre-training. In Table 1, we give information on RoBERTa model size, and in Table 2, we give information on the RoBERTa model pre-training dataset size.

APPENDIX D: Example of data

Other examples can be found at: https://asrs.arc.nasa.gov/search/dbol.html.

Narrative (written by the reporter):

Approximately 8 hours into our flight, my ears started to block. I swallowed to clear them, but it came back repeatedly. I spoke with 3 other flight attendants and they said they had the same symptoms. I called the cockpit and talked with the flight crew about the situation. They informed me that everything checked out all right. We were informed about a "PAC" being "out" during the Captain to crew, pre-flight briefing. I questioned flight crew if this had anything to do with our ears being blocked. Captain told me that the PAC that was out was like having a "spare tire." I questioned him because he informed the crew that the temperature in the cabin might be a problem. I asked him if the PAC situation had anything to do with air circulation or filtration, due to COVID transmittal. He said it was not going to affect the pressurization, air circulation or filtration. The ear blockage lasted for 15-20 minutes and didn't return the rest of the flight. Captain asked if we needed MedLink and we declined.

Synopsis (written by analysts)

Flight Attendant reported having ear blockage problems during flight and questioned if it had to do with one Pack being "out."

Aircraft related
Aircraft Operator : Air Carrier
Make Model Name : Commercial Fixed Wing
Crew Size.Number Of Crew : 2
Operating Under FAR Part : Part 121
Flight Plan : IFR
Mission : Passenger
Flight Phase : Cruise

Person related
Reference : 1
Location Of Person.Aircraft : X
Location In Aircraft : General Seating Area
Reporter Organization : Air Carrier
Function.Flight Attendant : Flight Attendant (On Duty)

Qualification.Flight Attendant : Current
ASRS Report Number.Accession Number : 1772104
Human Factors : Distraction
Human Factors : Physiological - Other

Events related
Anomaly.Aircraft Equipment Problem : Less Severe
Anomaly.Flight Deck / Cabin / Aircraft Event : Illness
Detector.Person : Flight Attendant
When Detected : In-flight
Result.General : None Reported / Taken

Assessments related
Contributing Factors / Situations : Aircraft
Primary Problem : Aircraft

APPENDIX E: Example of incident report from the 1987 to 2008 era

Narrative (written by the reporter):

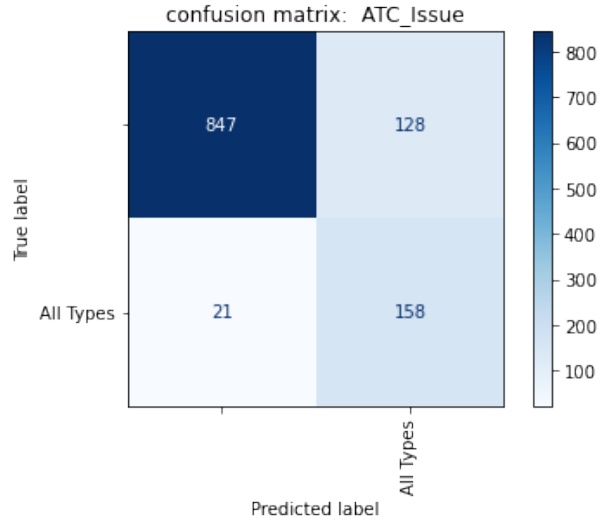
APCH CTL ISSUED ILS RWY 28R AND VECTORED US TO FINAL APCH CTLR. HE DSNDED US TO 3000 FT AND GAVE US AN INTERCEPT HDG AND ISSUED APCH CLRNC. I INTERCEPTED LOC AND TURNED INBOUND. THE CTLR SAID IT APPEARED WE WERE INTERCEPTING ILS RWY 28R WHICH WE WERE. HE INDICATED WE SHOULD BE ON APCH FOR ILS RWY 28L. HE THEY ISSUED US THE ILS FREQ 111.7 FOR ILS RWY 28R. WE ASKED HIM IF WE SHOULD BREAK OFF THE APCH AND HE ISSUED US A CLRNC FOR ILS RWY 28R THEN. WE WERE BEING VECTORED FOR R DOWNWIND WHICH IS USUAL FOR ILS RWY 28R, WHICH GAVE US MORE VERIFICATION FOR THE ILS RWY 28R. THE FINAL CTLR MAY HAVE THOUGHT WE WERE ISSUED AND EXPECTED RWY 28L, WHICH I DO NOT BELIEVE WE WERE. THERE WERE NO CONFLICTS. THERE WERE SNOW SQUALLS OVER THE AREA. I THINK MAYBE A MISUNDERSTANDING BTWN THE 2 APCH CTLRS DEVELOPED AS THERE WAS A HEARBACK AND READBACK FROM THEM EVERY TIME. THIS THING HAPPENS AND IN A HIGH TFC AREA WITH REDUCED VISIBILITY IT IS VERY IMPORTANT FOR THE CTLR AND PLT TO GET GOOD COMS ON IDENT OF THE RWY TO LAND ON. MAYBE A PROC FOR THE FINAL CTLR TO RENAME THE ILS AND GET A FINAL READBACK CLRNC FROM THE PLT PRIOR TO THE FINAL INTERCEPT HDG IS GIVEN, AND THE FINAL ILS CLRNC IS ISSUED. THE PLT WOULD THEN BE GIVEN A SECOND CHANCE TO CORRECT ANY ERROR THAT MIGHT EXIST.

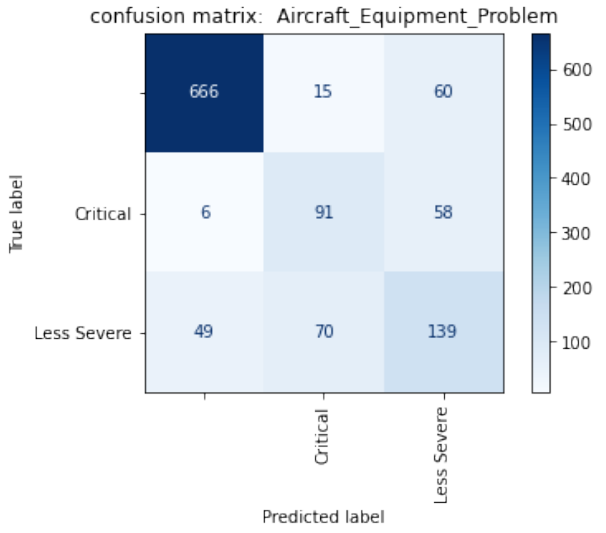
Synopsis (written by analysts)

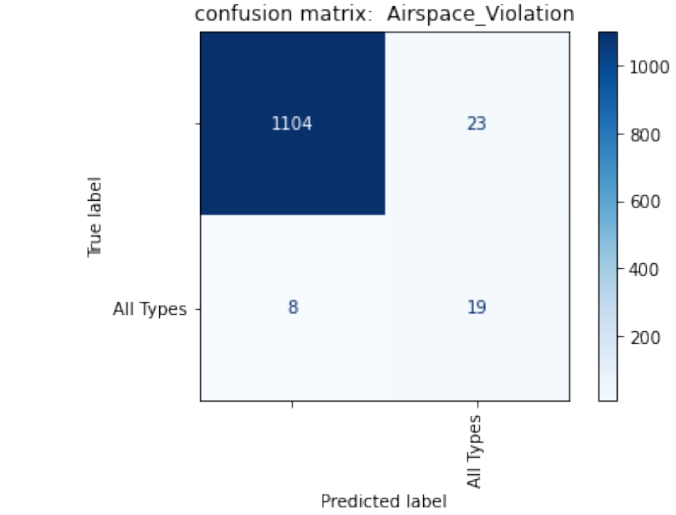
FLC OF A DC9-30 LINED UP WITH THE WRONG PARALLEL RWY RESULTING IN APCH CTLR INTERVENTION TO PROVIDE THEM WITH THE

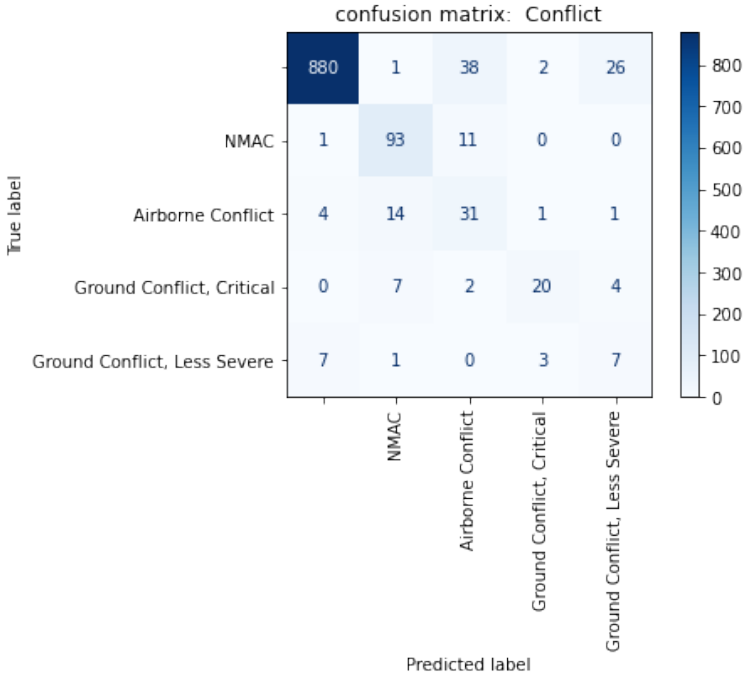
LOC FREQ FOR THE ASSIGNED PARALLEL RWY TO WHICH THEY BELIEVED THAT THEY WERE HEADED.

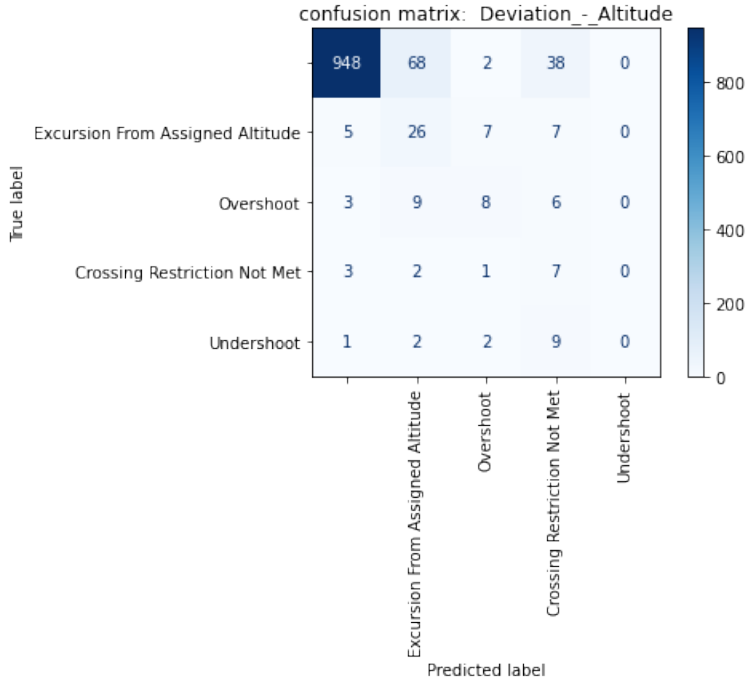
APPENDIX F: Confusion matrices

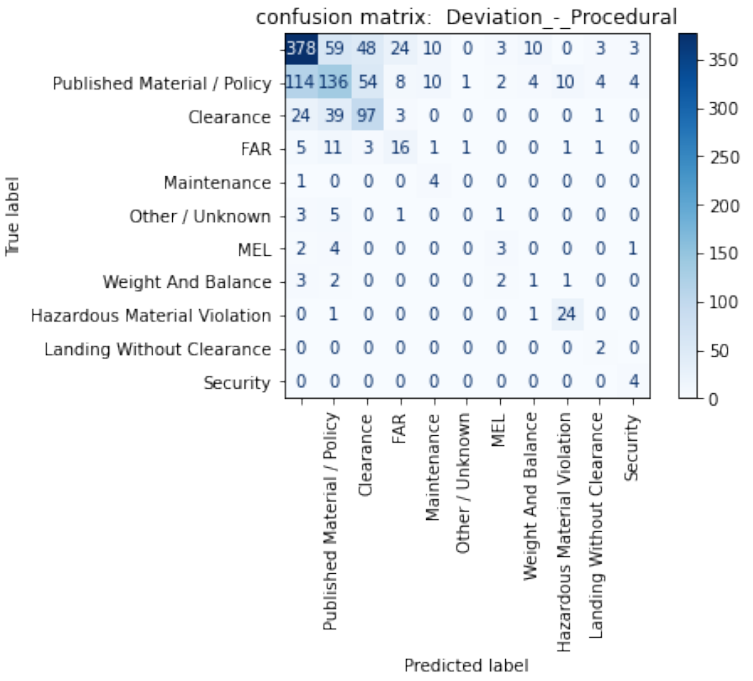


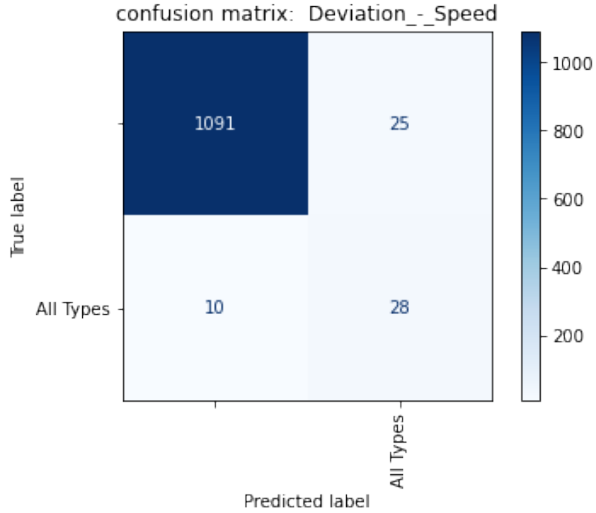


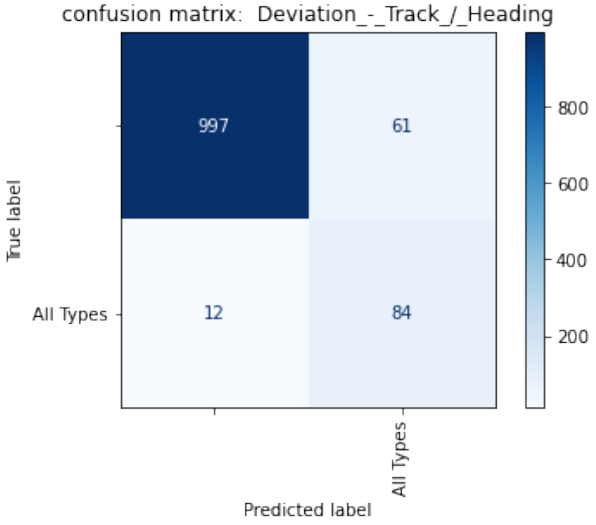

confusion matrix: Airspace_Violation
1104
23
8
19
All Types
True label
All Types
Predicted label
1000
800
600
400
200

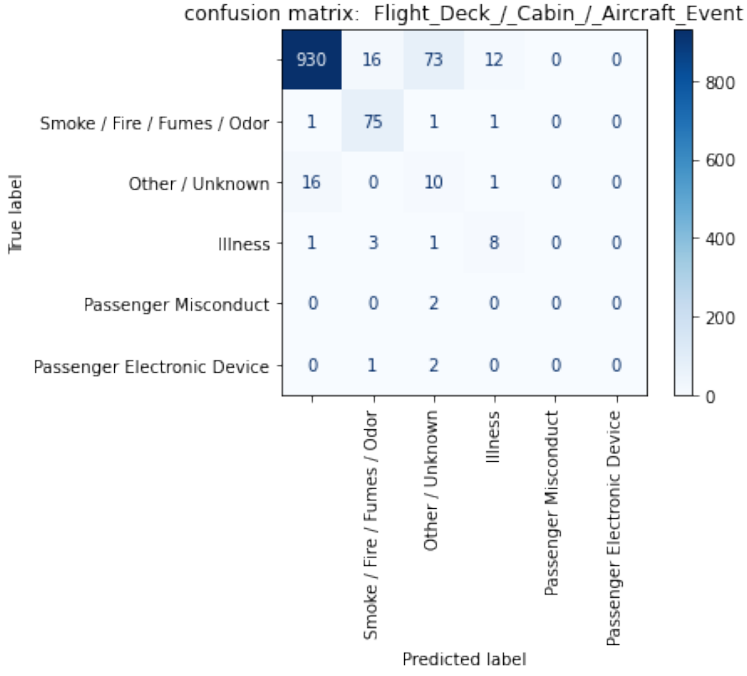

confusion matrix: Conflict
880 1 38 2 26
1 93 11 0 0
4 14 31 1 1
0 7 2 20 4
7 1 0 3 7
NMAC
Airborne Conflict
Ground Conflict, Critical
Ground Conflict, Less Severe
True label
NMAC
Airborne Conflict
Ground Conflict, Critical
Ground Conflict, Less Severe
Predicted label
800
700
600
500
400
300
200
100
0

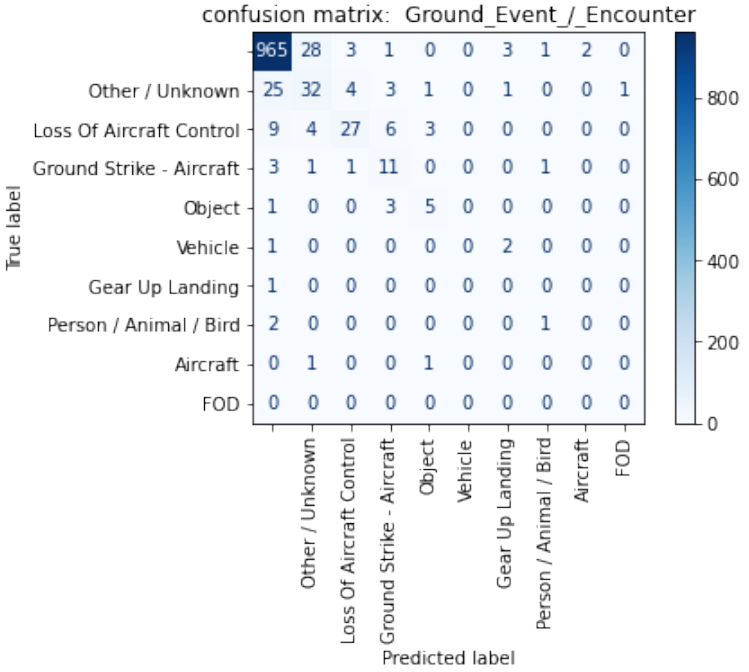

confusion matrix: Deviation_-_Altitude
948 68 2 38 0
5 26 7 7 0
3 9 8 6 0
3 2 1 7 0
1 2 2 9 0
Excursion From Assigned Altitude
Overshoot
Crossing Restriction Not Met
Undershoot
True label
Predicted label
800
600
400
200
0

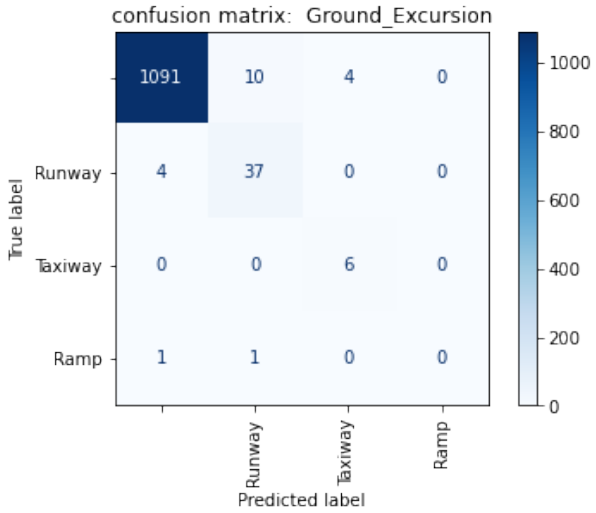

confusion matrix: Deviation_-_Procedural
378 59 48 24 10 0 3 10 0 3 3
114 136 54 8 10 1 2 4 10 4 4
24 39 97 3 0 0 0 0 0 1 0
5 11 3 16 1 1 0 0 1 1 0
1 0 0 0 4 0 0 0 0 0 0
3 5 0 1 0 0 1 0 0 0 0
2 4 0 0 0 0 3 0 0 0 1
3 2 0 0 0 0 2 1 1 0 0
0 1 0 0 0 0 0 1 24 0 0
0 0 0 0 0 0 0 0 0 2 0
0 0 0 0 0 0 0 0 0 0 4
Published Material / Policy
Clearance
FAR
Maintenance
Other / Unknown
MEL
Weight And Balance
Hazardous Material Violation
Landing Without Clearance
Security
True label
Predicted label
350
300
250
200
150
100
50
0

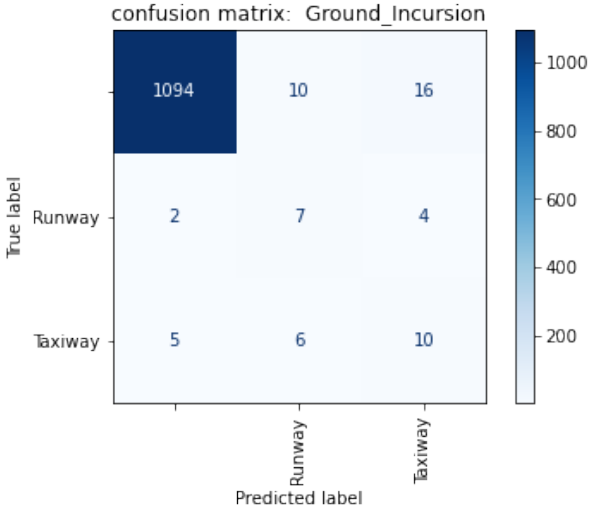

confusion matrix: Deviation_-_Speed
1091
25
10
28
True label
All Types
All Types
Predicted label
1000
800
600
400
200


confusion matrix: Deviation_-_Track_/_Heading
997
61
12
84
True label
All Types
All Types
Predicted label
800
600
400
200


confusion matrix: Flight_Deck_/_Cabin_/_Aircraft_Event
930 16 73 12 0 0
1 75 1 1 0 0
16 0 10 1 0 0
1 3 1 8 0 0
0 0 2 0 0 0
0 1 2 0 0 0
Smoke / Fire / Fumes / Odor
Other / Unknown
Illness
Passenger Misconduct
Passenger Electronic Device
True label
Predicted label
800
600
400
200
0


confusion matrix: Ground_Event_/_Encounter
965 28 3 1 0 0 3 1 2 0
25 32 4 3 1 0 1 0 0 1
9 4 27 6 3 0 0 0 0 0
3 1 1 11 0 0 0 1 0 0
1 0 0 3 5 0 0 0 0 0
1 0 0 0 0 0 2 0 0 0
1 0 0 0 0 0 0 0 0 0
2 0 0 0 0 0 0 1 0 0
0 1 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
Other / Unknown
Loss Of Aircraft Control
Ground Strike - Aircraft
Object
Vehicle
Gear Up Landing
Person / Animal / Bird
Aircraft
FOD
True label
Predicted label
800
600
400
200
0


confusion matrix: Ground_Excursion
1091
10
4
0
4
37
0
0
0
0
6
0
1
1
0
0
Runway
Taxiway
Ramp
True label
Runway
Taxiway
Ramp
Predicted label
1000
800
600
400
200
0


confusion matrix: Ground_Incursion
1094
10
16
2
7
4
5
6
10
Runway
Taxiway
True label
Runway
Taxiway
Predicted label
1000
800
600
400
200

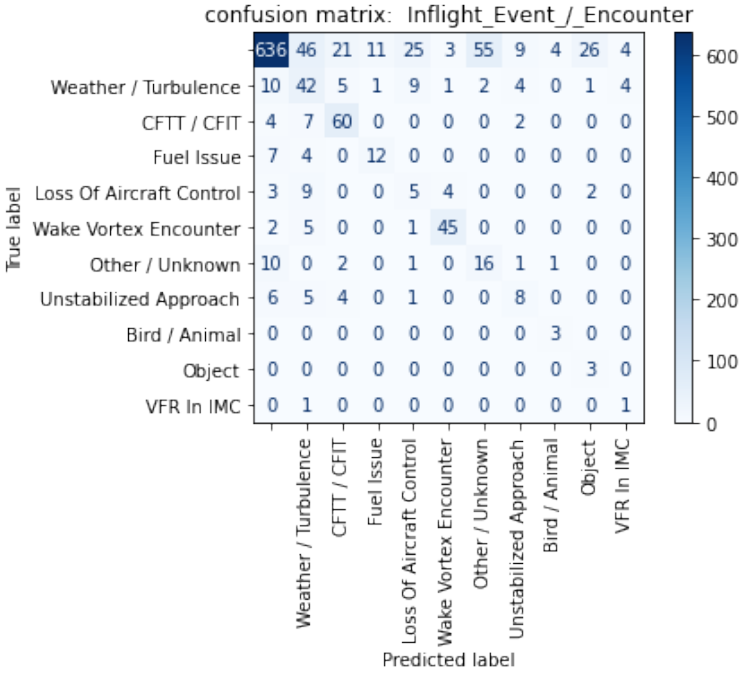


Figure 5: All the confusion matrices

APPENDIX G: Class composition of the multi-classification problems

Name	Composition	Shannon equitability index
Aircraft Equipment Problem	No (54.66%), Critical (24.39%), Less Severe (20.95%)	0.912
Conflict	No (85.32%), NMAC (6.2%), Airborne Conflict (4.5%), Ground Conflict, Critical (2.39%), Ground Conflict, Less Severe (1.59%)	0.374
Deviation - Altitude	No (91.2%), Excursion From Assigned Altitude (4.35%), Overshoot (2.37%), Crossing Restriction Not Met (1.62%), Undershoot (0.46%)	0.249
Deviation - Procedural	No (51.16%), Published Material / Policy (27.76%), Clearance (12.09%), FAR (4.01%), Maintenance (1.45%), Other / Unknown (1.29%), MEL (0.79%), Weight And Balance (0.63%), Hazardous Material Violation (0.34%), Landing Without Clearance (0.28%), Security (0.19%)	0.55
Flight Deck / Cabin / Aircraft Event	No (92.32%), Smoke / Fire / Fumes / Odor (3.51%), Other / Unknown (2.63%), Illness (1.25%), Passenger Misconduct (0.24%), Passenger Electronic Device (0.05%)	0.201
Ground Event / Encounter	No (91.03%), Other / Unknown (3.1%), Loss Of Aircraft Control (1.94%), Ground Strike - Aircraft (1.6%), Object (0.98%), Gear Up Landing (0.4%), Vehicle (0.4%), Person / Animal / Bird (0.23%), Aircraft (0.2%), FOD (0.12%)	0.2
Ground Excursion	No (96.89%), Runway (2.58%), Taxiway (0.47%), Ramp (0.06%)	0.112
Ground Incursion	No (96.38%), Runway (2.26%), Taxiway (1.35%)	0.163
Inflight Event / Encounter	No (76.97%), Weather / Turbulence (7.3%), CFTT / CFIT (3.07%), Fuel Issue (2.93%), Loss Of Aircraft Control (2.48%), Wake Vortex Encounter (2.17%), Other / Unknown (1.85%), Unstabilized Approach (1.83%), Bird / Animal (0.49%), Object (0.48%), VFR In IMC (0.43%)	0.417

Table 8: Anomaly subcategories that are multi-classification problems

APPENDIX H

Proof theorem 1 and closed formulas for $\hat{\Sigma}$ and Dg

Proof 1. *1. According to the vectorial central limit theorem, we have*

$$\sqrt{N}(\frac{1}{N}\sum_{n=0}^{N-1}(P_n - E[P_n]) \xrightarrow[N\to\infty]{\mathcal{L}} \mathcal{N}_{K^2}(\mathbf{0}\,, \Sigma)$$

Now setting $\sigma_1 = \sqrt{Dg^t \Sigma Dg}$, the Delta method ensures

$$\sqrt{N}((g(\hat{P}) - g(\frac{1}{N}\sum_{n=0}^{N-1} E[P_n])) \xrightarrow[N\to\infty]{\mathcal{L}} \mathcal{N}_1(0\,, \sigma_1^2)$$

2. According to the Slutsky Lemma

$$\frac{\sqrt{N}((g(\hat{P}) - g(\frac{1}{N}\sum_{n=0}^{N-1} E[P_n]))}{\hat{\sigma_1}} \xrightarrow[N\to\infty]{\mathcal{L}} \mathcal{N}_1(0\,, 1)$$

where,

$$\hat{\sigma_1} = \sqrt{Dg^t \hat{\Sigma} Dg}, \tag{13}$$

and $\hat{\Sigma}$ any consistant estimator of Σ.

Calculating $\hat{\Sigma}$

For elements in the diagonal of Σ, we have:

$$\begin{aligned} Cov(X_{l_1,m_1}, X_{l_1,m_1}) &= E(X_{l_1,m_1}^2) - E(X_{l_1,m_1})^2 \\ &= p_{l_1,m_1} \times (1 - p_{l_1,m_1}). \end{aligned} \tag{14}$$

We obtain that a consistent estimation for $\hat{\Sigma}$ is: $\hat{p}_{l_1,m_1} \times (1 - \hat{p}_{l_1,m_1})$.

For elements outside the diagonal, we obtain that:

$$\begin{aligned} Cov(X_{l_1,m_1,l_2,m_2}) &= E(X_{l_1,m_1} \times X_{l_2,m_2}) - E(X_{l_1,m_1}) \times E(X_{l_2,m_2}) \\ &= -p_{l_1,m_1} * p_{l_2,m_2}. \end{aligned} \tag{15}$$

We obtain that a consistent estimation for $\hat{\Sigma}$ is: $-\hat{p}_{l_1,m_1} * \hat{p}_{l_2,m_2}$.

Calculating Dg

We introduce the following notation for the purpose of clarity in our calculations: Let $\hat{p}_{i,.}$ designates the sum of the elements of row i of $\hat{P}$.
Let $\hat{p}_{.,j}$ designates the sum of the elements of column j of $\hat{P}$.
Let $\hat{p}_{i,-[j_0,\dots j_A]}$ with $A \in [0, K-1]$ designates $\hat{p}_{i,.} - \sum_{a=0}^{A} \hat{p}_{i,j_a}$
Let $\hat{p}_{-[i_0,\dots i_A],j}$ with $A \in [0, K-1]$ designates $\hat{p}_{.,j} - \sum_{a=0}^{A} \hat{p}_{i_a,j}$
Let $Tr(\hat{p})$ designates the sum of the elements in the diagonal of $\hat{P}$.

We must obtain $\frac{\partial g}{\partial \hat{p}_{i,j}}$. We distinguish between the case where $i = j$ and $i \neq j$.

When $i = j$, we have:

$$g(x_{l,l}) = \frac{x_{l,l} + a - (b + x_{l,l}^2 + x_{l,l} \times c)}{\sqrt{1 - d - (x_{l,l} + e)^2} \times \sqrt{1 - f - (x_{l,l} + g)^2}}. \tag{16}$$

Then we have:

$$\frac{\partial g}{\partial x_{l,l}} = \frac{(1 - 2 \times x_{l,l} - c) \times A \times B + ((x_{l,l} + e) \times \frac{B}{A} + (x_{l,l} + g) \times \frac{A}{B}) \times (x_{l,l} + a - (b + x_{l,l}^2 + x_{l,l} \times c))}{A^2 \times B^2}$$

with:

$$A = \sqrt{1 - d - (x_{l,l} + e)^2} \quad \text{and} \quad B = \sqrt{1 - f - (x_{l,l} + g)^2}.$$

When the input for g is our matrix $\hat{P}$, we have:

$$a = Tr(\hat{p}) - \hat{p}_{l,l}, \quad b = \sum_{k \neq l}^{K-1} (\hat{p}_{.,j} * \hat{p}_{i,.}) + \hat{p}_{-[l],l} \times \hat{p}_{l,-[l]}, \quad c = \hat{p}_{-[l],l} + \hat{p}_{l,-[l]}$$

$$d = \sum_{k \neq l}^{K-1} \hat{p}_{.,k}^2, \quad e = \hat{p}_{-[l],l}, \quad f = \sum_{k \neq l}^{K-1} \hat{p}_{k,.}^2$$

$$g = \hat{p}_{l,-[l]} \quad \text{and} \quad x_{l,l} = \hat{p}_{l,l}.$$

When $i \neq j$, we have:

$$g(x_{l,m}) = \frac{a_1 - (b_1 + (x_{l,m} + \hat{c}_1) \times d_1 + (x_{l,m} + e_1) \times f_1)}{\sqrt{1 - g_1 - (x_{l,m} + h_1)^2} \times \sqrt{1 - i_1 - (x_{l,m} + j_1)^2}}, \tag{17}$$

then we have:

$$\frac{\partial g}{\partial x_{l,m}} =$$

$$\frac{-(d_1 + f_1)A_1B_1 + ((x_{l,m} + h_1)B_1/A_1 + (x_{l,m} + j_1)A_1/B_1)(a_1 - (b_1 + (x_{l,m} + \hat{c}_1)d_1 + (x_{l,m} + e_1)f_1)}{A_1^2 B_1^2}$$

with:

$$A_1 = \sqrt{1 - g_1 - (x_{l,m} + h_1)^2} \quad \text{and} \quad B_1 = \sqrt{1 - i_1 - (x_{l,m} + j_1)^2}.$$

When the input for g is our matrix $\hat{P}$, we have:

$$a_1 = Tr(\hat{p}), \quad b_1 = \sum_{k \neq (l,m)}^{K-1} (\hat{p}_{.,k} * \hat{p}_{k,.}), \quad c_1 = \hat{p}_{l,-[m]}$$

$$d_1 = \hat{p}_{.,l}, \quad e_1 = \hat{p}_{-[l],m}, \quad f_1 = \hat{p}_{m,.}$$

$$g_1 = \sum_{k \neq m}^{K-1} \hat{p}_{.,k}^2, \quad h_1 = \hat{p}_{-[l],m}, \quad i_1 = \sum_{k \neq l}^{K-1} \hat{p}_{k,.}^2$$

$$j_1 = \hat{p}_{l,-[m]} \quad \text{and} \quad x_{l,m} = \hat{p}_{l,m}.$$

Proof theorem 2 and closed formulas for Σ_a and DJ

Proof 2. *According to the vectorial central limit theorem*

$$\sqrt{N}(\frac{1}{N}\sum_{n=0}^{N-1}(U_n - E[U_n]) \xrightarrow[N\to\infty]{\mathcal{L}} \mathcal{N}_{2K^2+1}(\mathbf{0}, \Sigma_a),$$

where $\Sigma_a = Cov(U_n)$ is a covariance matrix of size $2K^2+1$. According to the delta method

$$\sqrt{N}((J(\overline{U}_N) - J(\frac{1}{N}\sum_{n=0}^{N-1}E[U_n])) \xrightarrow[N\to\infty]{\mathcal{L}} \mathcal{N}_2(\mathbf{0}, \Sigma_b).$$

where $\Sigma_b = DJ^t\Sigma_a DJ$. Now since $J(\overline{U}_N) = (M\hat{C}C_1, M\hat{C}C_2)$ we have

$$\sqrt{N}(\begin{bmatrix} M\hat{C}C_1 \\ M\hat{C}C_2 \end{bmatrix} - \begin{bmatrix} MCC_{1true} \\ MCC_{2true} \end{bmatrix}) \xrightarrow[N\to\infty]{L} \mathcal{N}_2(\mathbf{0}, \Sigma_b).$$

Applying again the μDelta method we get

$$\sqrt{N}((M\hat{C}C_1 - M\hat{C}C_2) - (MCC_{1true} - MCC_{2true}) \xrightarrow[N\to\infty]{\mathcal{L}} \mathcal{N}(0, \sigma_2^2).$$

where $\sigma_2 = \sqrt{Df^t\Sigma_b Df}$, with Df the gradient of the application f
Under H_0, we have

$$\sqrt{N}(M\hat{C}C_1 - M\hat{C}C_2) \xrightarrow[N\to\infty]{\mathcal{L}} \mathcal{N}(0, \sigma_2^2).$$

calculating $\hat{\Sigma_a}$

We have 9 cases:

- $Cov(X1_{l_1,m_1}, X1_{l_2,m_2})$
- $Cov(X1_{l_1,m_1}, X1_{l_1,m_1})$
- $Cov(X2_{l_1,m_1}, X2_{l_2,m_2})$
- $Cov(X2_{l_1,m_1}, X2_{l_1,m_1})$
- $Cov(X1_{l_1,m_1}, X2_{l_1,m_1})$
- $Cov(X1_{l_1,m_1}, X2_{l_2,m_2})$
- $Cov(X1_{l_1,m_1}, \mathbb{1}_{\delta_n=1})$
- $Cov(X2_{l_1,m_1}, \mathbb{1}_{\delta_n=1})$
- $Cov(\mathbb{1}_{\delta_n=1}, \mathbb{1}_{\delta_n=1})$

with $(l_1, m_1) \neq (l_2, m_2)$.

The four first cases are already covered by our previous work on confidence intervals.
For the two next cases, we obtain that:

$$\begin{aligned} Cov(X1_{l_1,m_1}, X2_{l_2,m_2}) &= E(X1_{l_1,m_1} \times X2_{l_2,m_2}) - E(X1_{l_1,m_1}) \times E(X2_{l_2,m_2}) \\ &= -q1_{l_1,m_1} * q2_{l_2,m_2}. \end{aligned} \tag{18}$$

We obtain that a consistent estimation for $\hat{\Sigma_a}$ is: $-\hat{q1}_{l_1,m_1} * \hat{q2}_{l_2,m_2}$.

$$\begin{aligned} Cov(X1_{l_1,m_1}, X2_{l_1,m_1}) &= E(X1_{l_1,m_1} \times X2_{l_1,m_1}) - E(X1_{l_1,m_1}) \times E(X2_{l_1,m_1}) \\ &= -q1_{l_1,m_1} q2_{l_1,m_1}. \end{aligned} \tag{19}$$

We obtain that a consistent estimation for $\hat{\Sigma_a}$ is: $-\hat{q1}_{l_1,m_1}\hat{q2}_{l_1,m_1}$.

For the last three cases, because we have:

$$E(X1_{l_1,m_1} \times \mathbb{1}_{\delta_n=1}) = E(X_{l_1,m_1} \times (1 - \mathbb{1}_{\delta_n=1}) \times \mathbb{1}_{\delta_n=1}) = 0,$$

$$E(X2_{l_1,m_1} \times \mathbb{1}_{\delta_n=1}) = E(X_{l_1,m_1} \times \mathbb{1}_{\delta_n=1} \times \mathbb{1}_{\delta_n=1}) = q2_{l_1,m_1},$$

$$E(\mathbb{1}_{\delta_n=1}) = \lambda,$$

we can deduce that:

$$Cov(X1_{l_1,m_1}, \mathbb{1}_{\delta_n=1}) = -\lambda \times q1_{l_1,m_1}, \tag{20}$$

We obtain that a consistent estimation for $\hat{\Sigma_a}$ is: $-\hat{\lambda} \times \hat{q1}_{l_1,m_1}$.

$$Cov(X2_{l_1,m_1}, \mathbb{1}_{\delta_n=1}) = q2_{l_1,m_1}(1 - \lambda), \tag{21}$$

We obtain that a consistent estimation for $\hat{\Sigma_a}$ is: $\hat{q2}_{l_1,m_1}(1 - \hat{\lambda})$.

$$Cov(\mathbb{1}_{\delta_n=1}, \mathbb{1}_{\delta_n=1}) = \lambda(1 - \lambda). \tag{22}$$

We obtain that a consistent estimation for $\hat{\Sigma_a}$ is: $\hat{\lambda}(1 - \hat{\lambda})$.

Calculating DJ

We obtain:

$$\frac{\partial DJ_1(x,y,z)}{\partial y} = \frac{\partial g(\frac{x}{1-z}))}{\partial y} = 0, \quad \frac{\partial DJ_2(x,y,z)}{\partial x} = \frac{\partial g(\frac{y}{z})}{\partial x} = 0$$

Conversely, we have that:

$$\frac{\partial DJ_1(x,y,z)}{\partial x} = \frac{1}{1-z}\frac{\partial g(\frac{x}{1-z})}{\partial x}, \quad \frac{\partial DJ_2(x,y,z)}{\partial y} = \frac{1}{z}\frac{\partial g(\frac{y}{z})}{\partial y}$$

To calculate $\frac{\partial DJ_1(x,y,z)}{\partial z}$ and $\frac{\partial DJ_2(x,y,z)}{\partial z}$, we introduce the following:

$$\hat{\lambda} = \frac{1}{N}\sum_{n=0}^{N-1} \mathbb{1}_{\delta_n=1}, \tag{23}$$

$$\hat{q1}_{i,j} = \frac{1}{N}\sum_{n=0}^{N-1} X1_n(i,j), \tag{24}$$

$$\hat{q2}_{i,j} = \frac{1}{N}\sum_{n=0}^{N-1} X2_n(i,j). \tag{25}$$

We have:

$$M\hat{C}C_1 = \frac{(\frac{1}{1-\hat{\lambda}})a_2 - (\frac{1}{1-\hat{\lambda}})^2 b_2}{\sqrt{1 - (\frac{1}{1-\hat{\lambda}})^2 c_2} \times \sqrt{1 - (\frac{1}{1-\hat{\lambda}})^2 d_2}} \tag{26}$$

with:

$$a_2 = Tr((\hat{q1}_{i,j})), \quad b_2 = \sum_{k}^{K-1}(\hat{q1}_{.,k} * \hat{q1}_{k,.}), \quad c_2 = \sum_{k}^{K-1} \hat{q1}^2_{.,k} \quad \text{and} \quad d_2 = \sum_{k}^{K-1} \hat{q1}^2_{k,.}$$

We note:

$$C2(z) = \sqrt{1 - (\frac{1}{1-z})^2 c_2} \quad \text{and} \quad D2(z) = \sqrt{1 - (\frac{1}{1-z})^2 d_2}.$$

Then we have:

$$\frac{\partial DJ_1((\hat{q1}_{i,j}), (\hat{q2}_{i,j}), \hat{\lambda})}{\partial z} = (\frac{1}{1-\hat{\lambda}})^2 (\frac{a_2 - 2(\frac{1}{1-\hat{\lambda}})b_2}{C2(\hat{\lambda})D2(\hat{\lambda})} + \frac{(\frac{1}{1-\hat{\lambda}})(C2(\hat{\lambda})^2 d_2 + D2(\hat{\lambda})^2 c_2)((\frac{1}{1-\hat{\lambda}})a_2 - (\frac{1}{1-\hat{\lambda}})^2 b_2)}{(C2(\hat{\lambda})D2(\hat{\lambda}))^3})$$

Respectively, we have:

$$M\hat{C}C_2 = \frac{(\frac{1}{\hat{\lambda}})a_3 - (\frac{1}{\hat{\lambda}})^2 b_3}{\sqrt{1 - (\frac{1}{\hat{\lambda}})^2 c_3} \times \sqrt{1 - (\frac{1}{\hat{\lambda}})^2 d_3}} \tag{27}$$

with:

$$a_3 = Tr((\hat{q2}_{i,j})), \quad b_3 = \sum_{k}^{K-1}(\hat{q2}_{.,k} * \hat{q2}_{k,.}), \quad c_3 = \sum_{k}^{K-1} \hat{q2}^2_{.,k} \quad \text{and} \quad d_3 = \sum_{k}^{K-1} \hat{q2}^2_{k,.}$$

We note:

$$C3(z) = \sqrt{1 - (\frac{1}{z})^2 c_3} \quad \text{and} \quad D3(z) = \sqrt{1 - (\frac{1}{z})^2 d_3}.$$

Then we have:

$$\frac{\partial DJ_2((\hat{q1}_{i,j}), (\hat{q2}_{i,j}), \hat{\lambda})}{\partial z} = -(\frac{1}{\hat{\lambda}})^2 (\frac{a_3 - 2(\frac{1}{\hat{\lambda}})b_3}{C3(\hat{\lambda})D3(\hat{\lambda})} + \frac{(\frac{1}{\hat{\lambda}})(C3(\hat{\lambda})^2 d_3 + D3(\hat{\lambda})^2 c_3)((\frac{1}{\hat{\lambda}})a_3 - (\frac{1}{\hat{\lambda}})^2 b_3)}{(C3(\hat{\lambda})D3(\hat{\lambda}))^3})$$

APPENDIX I: Summary of the NLP pipeline

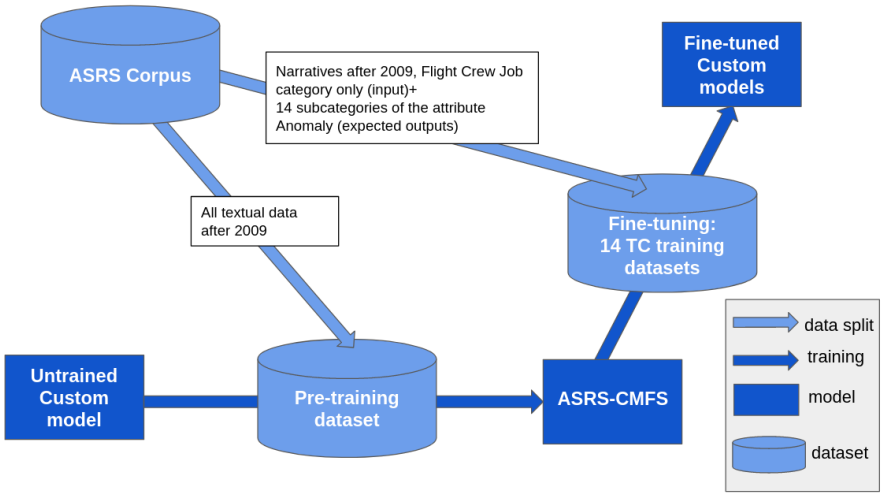


Figure 6: NLP pipeline

We created and pre-trained from scratch a custom transformer-based model on the language modeling task, using part of the ASRS corpus for that purpose, as detailed in the Sections 2.1.2 and 2.1.3. This pre-trained custom model was then trained separately on 14 different text classification training datasets that were also extracted from the ASRS corpus, with each classification problem corresponding to a subcategory of the Anomaly attribute. For each subcategories, the labels can be seen in Tables 8 and 4. The choice of fine tuning data is detailed in Section 2.2.3.
Once the pre-trained model was trained on our fine tuning data, we evaluated it on the testing datasets, as well as gauged the impact of sliding windows on our model's performance. Details on the metric used for evaluations can be found in Section 3, which is dedicated to the theoretical and statistical study of the efficiency of the classification procedure.