Transformer-based model on aviation incident reports

Samuel$_1$ Kierszbaum$_1$[*] — Laurent$_2$ Lapasset$_2$[**] — Thierry$_3$ Klein$_3$[***]

[*] *ENAC*
[**] *ENAC*
[***] *ENAC*

RÉSUMÉ. *Les modèles transformers sont plus performants que les humains sur plusieurs tâches de compréhension de langage comme la classification de texte, et sont utilisés dans des domaines spécialisés comme la médecine. Dans ce contexte, notre objectif est d'explorer l'utilisation de ces modèles afin d'aider les analystes impliqués dans la sûreté en aviation, dans leur travail sur les rapports d'incidents. Dans cet article, on travaille avec le data set Aviation Safety Reporting System (ASRS), contenant des rapports d'incidents en anglais ainsi que des métadonnées. Ces rapports sont caractérisés par l'utilisation de vocabulaire, abbréviations et de langage spécifique au domaine de l'aviation. Cette caractéristique rend leur analyse difficile. Nous explorons l'idée que le travail des analystes peut-être reformulé en tâches de compréhension de langage, en respectant certaines conditions. Nous proposons ensuite une approche expérimentale ou l'on utilise un modèle transformer sur l'une de ces tâches.*

ABSTRACT. *Recently, transformer-based models have beaten humans in Natural Language Understanding (NLU) tasks such as text classification, and have been used in specialized fields such as healthcare. In this context, our general aim is to explore how such models could help support analysts working in safety in aviation, in particular when they are used on incident reports. In this article, we work with the Aviation Safety Reporting System (ASRS) data set. It is made up of incident reports in English, as well as supporting metadata. Such reports are characterized by the heavy use of specialized language, abbreviations, and domain-specific vocabulary, as opposed to general day-to-day English. We explore the idea that analyst work can be re-framed as a set of NLU tasks. We then propose an experimental procedure to try and use transformer-based models on one of these tasks.*

MOTS-CLÉS : *NLP$_1$, ASRS$_2$, BERT$_3$.*

KEYWORDS: *NLP$_1$, ASRS$_2$, BERT$_3$.*

1. Introduction

Our interest in this article is with incorporating text-related technology to Safety in Aviation. In this context, our general aim is to find ways to leverage transformer-based algorithms to help support analysts that work in this domain.

This new generation of algorithms has beaten humans in many Natural Language Processing tasks (NLP) (Wang *et al.*, 2019 ; Wang *et al.*, 2020). The publication of the Transformer article (Vaswani *et al.*, 2017), from which current state of the art models are inspired is fairly recent, and showcases the attention mechanism used by transformer-based models to get state of the art results.

One can arguably compare the attention mechanism in transformers to how humans treat textual information. When we read a sentence, we don't treat each word as a separate unit but rather look at the sentence as a whole to provide nuance for the meaning of each word.

Real-life applications of such models exist in fields that use specialized language such as the legal field (Chalkidis *et al.*, 2020), scientific field (Beltagy *et al.*, 2019) or medical field (Gu *et al.*, 2021).

In this article, we explore how the work done by analysts at the National Aeronautics and Space Administration (NASA), to maintain the Aviation Safety Reporting System (ASRS) data set, can be re-framed as a set of NLP tasks (Tulechki, 2015).

To the best of our knowledge, our article presents these original contributions:

– Emphasizing importance of ensuring that algorithms are working in conditions that are as close as possible to the ones of ASRS safety analysts, and proposing a set of guidelines with this aim in mind.

– Initiating the work of exploring the use of transformer-based models in this context on a particular task and model.

2. Context

2.1. *Data set*

The ASRS is a semi-structured data set, containing voluntarily reported information about incident occurrences or perceived dangerous situations in an aviation context. These reports are referred to as "narratives". The data set also contains short summaries of these reported incidents, the so-called "synopses", along with supporting metadata. For occurrences where the analyst contacted the reporter for further details on the incident, there is also a "callback", containing the additional information.

The ASRS was created in April 1976 (ASRS, 2019). Since its creation, the report intake of ASRS has been growing exponentially, with an average of 2 248 reports per

week, at the end of 2 019 (ASRS, 2019). In this context, “the need to automatically classify reports in a given taxonomy” has already been identified in previous work (Tulechki, 2015).

Upon reception, analysts initially identify reports that deserve entry into the public ASRS data set. Before entering into the ASRS data set, “reports are codified using the ASRS taxonomy” (ASRS, 2019), de-identified, corrected, and potentially “an ASRS analyst may choose to call a reporter on the telephone to clarify any information the reporter provided” (ASRS, 2019). Given that we are working with the public version of the ASRS data set, some of this initial work is considered out of scope in our article. In the rest of the article, the “analyst work” that we suggest can be re-framed as NLP tasks, refers to the following:

– codifying the reports, by adding analyst-produced metadata using a pre-defined taxonomy

– writing the synopses

Identified role of metadata includes helping to query the data set, monitoring trends, and producing KPIs (Tulechki, 2015). The ASRS corpus is partly independently coded (Tulechki, 2015), meaning that some of the metadata is produced by the reporter, while the rest is produced by safety analysts. To identify the latter, we looked at the metadata categories present in the data set but not in the reporting forms. When looking at the reporting forms, we can further notice that some metadata are specific to the reporter’s job (ASRS, 2021).

The authors received the ASRS public data set on a disk. Below is a description of the dataset as received.

The occurrences range from 1987 to 2019. We distinguish between three kinds of documents in the data set. The narratives, the synopses, and the callbacks. The narrative is written by the reporter. The synopsis and the callback are written by an analyst. All of these texts make use of elements of language that are specific to the aviation domain. This characteristic is referred to as “in-domain”. It stands in contrast with the “out-domain” text that uses English in other contexts than aviation. A feature of our textual data is the style, which has changed with time. We observe that for texts before 2009 excluded, the text is written in all upper case letters, there is heavy use of both standardized and not standardized abbreviations and English is not grammatically correct. It is not the case for reports after the year 2009 included, with the use of lower case and upper case letters, standard abbreviations, and correct English. A sample of the ASRS data set can be found in the appendix.

documents	385 492
size	287MB
Space-delimited word count	50 204 970

Tableau 1 – Amount of textual data in the working corpus

2.2. *Pre-training and fine-tuning*

The main reason for the success of transformer-based models might be their ability to leverage a massive amount of unlabelled textual data, to learn to model a language using the attention mechanism. It is the so-called Language Modeling task. This step of training a model on a prior task to improve performance on a downstream task is called pre-training. Fine-tuning designates the step where a pre-trained model is once again trained, generally on a supervised task with fewer data. The tasks on which a pre-trained model is fine-tuned are the "downstream tasks". Models that are pre-trained obtain better results. The concept behind this increase in performance is called "transfer learning". The idea is that through learning on a pre-training task, the algorithm gains transverse skills that can increase performance on downstream tasks. Depending on the specifics of the transformer-based model architecture and training protocol, the implementation details of these two steps are not the same, but the idea remains.

According to (Chalkidis *et al.*, 2020 ; Gu *et al.*, 2021 ; Beltagy *et al.*, 2019), when used in a context where the text is related to a specialized field, a heavy performance factor of language models is how the pre-training incorporates in-domain data.

We distinguish between two main strategies when using in-domain data in the pre-training step. The first strategy is doing additional pre-training with in-domain data on an already pre-trained model (on general English data). We refer to this strategy as "mixed-domain pre-training". The other strategy consists in training a yet untrained model from scratch on only in-domain data. We refer to this strategy as "pre-training from scratch".

3. Re-framing analyst work as NLP tasks

3.1. *Guidelines*

One of our main concerns is to use our algorithms in field conditions that are as close as possible to the analysts'. We work under the assumption that performance in real-life situations is what gives our algorithms value.
In particular, we propose the following set of guidelines:

– For an occurrence, algorithms, not unlike analysts, should only use the reporter-provided information (both metadata and textual) and not other analyst-produced metadata as input when making a prediction. It stands in contrast with previous work (Zhang et Mahadevan, 2019).

– All of the occurrences used in the training data set should have happened before all of the occurrences in the test data set. In real working conditions, the analyst works on "new" reports, with the possibility of yet-unseen novelty in the incident circumstances.

– All of the occurrences used for both training and testing should be from after 2009 because analysts currently work on reports using this kind of style.

These working assumptions give us initial guidelines on how to constitute our training and testing data sets.

3.2. *Type of tasks*

As previously mentioned, our focus is on re-framing any of the following analyst work as NLP tasks: codifying of the reports and writing of the synopses.

Re-framing the writing of the synopses is straightforward. It can be seen as an Abstractive Text Summarization task, where the algorithm generates a summary of the input, using sentences that may not be in the original text.

For the act of codifying the reports, we work under the assumption that one can re-frame this work as either a variation of the text classification task or an Extractive Question Answering task.

Extractive Question Answering algorithms work in the following fashion: given an input question and an input text, the algorithm provides an answer to the input question under the form of a subsequence of the input text (if it exists). This is useful for the case where the reporter has to extract information directly from the text. For instance, in the case of the metadata "Aircraft component", the algorithm would extract from the text all mentions of aircraft components that would have been involved in an incident. The input question would be: "What were the aircraft components involved?"

Text classification is the task of assigning a piece of text to an appropriate category. For instance, the metadata "Primary Problem" is an assessment made by an ASRS analyst of the main factor in an incident. He has to choose between 17 categories (weather, airspace structure, etc.). There can also be cases where the metadata is multi-label, a variation of the classification task where more than one label can be assigned to each instance. It is the case for the "Human factors" metadata. There can be multiple human factors involved in a single incident (fatigue, workload, etc.).

In this article, we only worked on one of the metadata. We leave for future work the re-framing of codifying the other metadata as NLP tasks.

4. Operational choices

Because of limited computational resources and time, we have used only one kind of model and one task when doing our experiments. We give further information on these choices below.

4.1. *Choice of model*

We have chosen the Roberta model. It is well-positioned in both Glue and Superglue benchmarks leader-boards (Wang *et al.*, 2019 ; Wang *et al.*, 2020).

We use three variations of the model. One is the plain pre-trained Roberta-base (Liu *et al.*, 2019). One is the same pre-trained Roberta-based further pre-trained on ASRS data. The last is an untrained model trained from scratch on only ASRS data with the tokenizer also trained from scratch.

For fairness, all models have the same configuration. The second and third models are pre-trained using the same set of hyper-parameters and training data, as seen in the table below:

Epochs	3
Learning rate	5E-5
Warm-up ratio	0.06
Batch size	8
Training data	all ASRS text

Tableau 2 – Hyper-parameters pre-training

We trained all the models using GeForce RTX 2080 Ti GPUs.

For the textual data before 2009, we used 37 000 feet website ASRS reports (Kuo, 2019). They are partially converted to match current reports writing style with disabbreviation of some words and use of both upper and lower case.

We used simpletransformers (Rajapakse, 2019), Huggingface's Transformer and Tokenizer for model instantiating and training (Wolf *et al.*, 2020).

Ideally, we would have liked to use the same set of hyper-parameters for pre-training from scratch and mixed-domain pre-training as in the original model (Liu *et al.*, 2019). This was unfortunately impossible due to the lack of time and computational resources (with all of our GPUs, it would have taken at a minimum more than a month to do one pre-training run). Another caveat we must mention is that our training data set size is small when compared with the data set for the original model or what was used in other articles dealing with specialized data (Liu *et al.*, 2019 ; Gu *et al.*, 2021 ; Chalkidis *et al.*, 2020). More data equates better results when training transformer-based model (Raffel *et al.*, 2020). This is the reason why we chose to incorporate pre-2009 data, even if the style difference might have had a negative learning effect.

We leave for future work exploring the use of other hyper-parameters and training data for pre-training.

4.2. *Choice of task for fine-tuning*

For the fine-tuning task, we have chosen to predict the analyst-produced metadata "Events Anomaly ATC". It is a binary classification task that is fairly unbalanced. We don't have a particular reason for choosing this task among the others. We leave for future work to investigate the other tasks. We have constituted the training and testing data set following the guidelines from the "guidelines" section, with the test data set being made from data of the year 2019, and the training data set made from data between 2009 and 2019 excluded. Furthermore, we have only used data from reporters corresponding to the "Flight Crew" job category. This is because they are the most important source of reports and because we work under the assumption that the data is too heterogeneous depending on the job of the reporter. This impression is reinforced by the use of different reporting forms depending on the job of the reporter.

All of our three pre-trained models are fine-tuned and compared on this task with the same set of hyper-parameters:

Epochs	3
Learning rate	6E-5
Gradient accumulation steps	4
Warm-up ratio	0.06
Batch size	32
sliding window	True

Tableau 3 – Hyper-parameters fine-tuning

We chose hyperparameters close to the ones used in the Roberta (Liu *et al.*, 2019) and Bert (Devlin *et al.*, 2019) paper for finetuning on Glue (Wang *et al.*, 2019). Our goal was to compare the models with each other on an equal footing, as opposed to finding the best hyperparameters for each of them.

For reports that were longer than 512 tokens, we used the sliding windows function provided by simpletransformers.

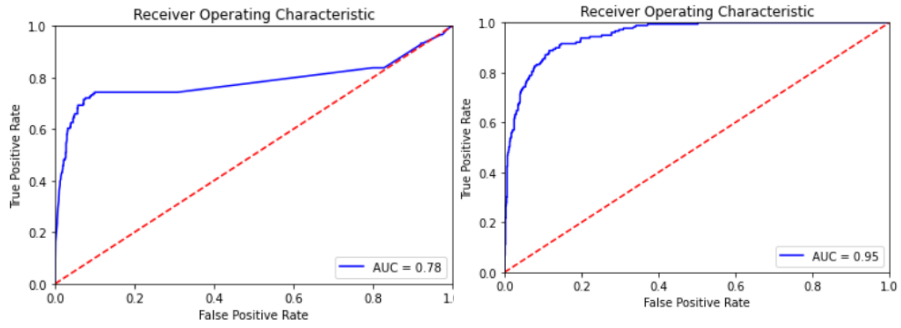
5. Result

We give the results on the evaluation data set in the table below, where tp stands for true positive, fp for false positive, tn for true negative, fn for false negative, and mcc for Matthews correlation coefficient. We notice that the model trained from scratch always predict the majority class (negative class). We hypothesize that this poor performance is due to either lack of pre-training time or lack of training data.

When we look at the ROC curves in figure 1, we notice that for a higher False Positive Rate (fp/tn+fp), the corresponding True Positive Rate (tp/tp+fn) for the mixed model is lower. The regular model surprisingly has the upper hand.

Model	tp	tn	fp	fn	mcc	precision	recall	f1	AUC
Regular	136	915	60	43	0.67	70.2	77.65	73.74	0.95
Mixed	109	934	41	70	0.61	72.15	63.69	67.66	0.78
Scratch	0	975	0	179	0	-	-	-	-

Tableau 4 – Results on the evaluation set of our various models



(a) Mixed model

(b) Regular model

Figure 1 – ROC curve and AUC

6. Discussion

6.1. *Limitations*

Because of the computational resource and time constraints, we have not been able to properly investigate how to obtain the best score possible on our task. We would also have liked to explore other tasks. Our work also lacks a qualitative analysis, based on ASRS safety experts knowledge, to assess what could have been the reason for our wrong predictions. We leave this for future work.

6.2. *Related work*

The domain of NLP applied to aviation safety has produced many articles (Zhang et Mahadevan, 2019 ; Tanguy *et al.*, 2016 ; Kuhn, 2018 ; Robinson, 2018 ; Shi *et al.*, 2017). Among them, those which use the ASRS dataset are not uncommon. In Zhang et Mahadevan (2019), authors use a hybrid model to quantify the risk associated with an occurrence. In Kuhn (2018), the author uses topic modeling to identify trends and topics. In Robinson (2018), authors use latent semantic analysis to try to predict the “Primary Problem” as well as “contributing factors” metadata in incidents.

6.3. *Future work*

The present article is an initial attempt at using transformer-based models for supporting analysts in their work. There is still much work to be done. We give here a few axes of development.

– We have started gathering data on what is the thought process of safety analysts during analysis, with the intent to create better models as well as improving their qualitative assessments. We will also investigate how this know-how can be used to do domain-specific knowledge integration, feature engineering on the data, and finding new mechanisms to adapt our models specifically to the problems at hand.

– We want to keep exploring other transformer-based models and all of the tasks that can be made out of the ASRS data set using the above guidelines. For instance, the production of synopses could be interesting to tackle. Also, as the field of NLP is evolving rapidly, there are faster and more accurate models coming.

– On a higher level, we are interested in refining the scope of utility of transformer-based models. This axis of development investigates how to get the best-added value when using transformer-based models in the context of Safety in Aviation. Our goal when working in a field such as Aviation is not to get a better score on a benchmark, but to increase Safety. We must not lose track of this consideration when working in this field. It requires communication between the workers on the NLP side and the workers on the Safety side, to uncover use cases that can be tackled by NLP models.

7. Conclusion

We can conclude from this work that transformer-based models can potentially be used to do some of the analyst work. We proposed a set of guidelines on how to convert that work into NLP tasks, with the intent to prioritize the end-user needs in mind. We also showed a limited implementation of using a transformer-based model on one of these tasks.
We do not know yet if these models are powerful enough to attain satisfying results. We feel that this work gives directions towards where efforts are needed.

8. Bibliographie

ASRS, “ASRS Program Briefing”, 2019.

ASRS, “ASRS reporting form”, 2021.

Beltagy I., Lo K., Cohan A., “SciBERT: A Pretrained Language Model for Scientific Text”, 2019.

Chalkidis I., Fergadiotis M., Malakasiotis P., Aletras N., Androutsopoulos I., “LEGAL-BERT: The Muppets straight out of Law School”, 2020.

Devlin J., Chang M.-W., Lee K., Toutanova K., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, 2019.

Gu Y., Tinn R., Cheng H., Lucas M., Usuyama N., Liu X., Naumann T., Gao J., Poon H., “Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing”, 2021.

Kuhn K., “Using structural topic modeling to identify latent topics and trends in aviation incident reports”, *Transportation Research Part C: Emerging Technologies*, vol. 87, p. 105-122, 02, 2018.

Kuo S., “37000 Feet”, 2019.

Liu Y., Ott M., Goyal N., Du J., Joshi M., Chen D., Levy O., Lewis M., Zettlemoyer L., Stoyanov V., “RoBERTa: A Robustly Optimized BERT Pretraining Approach”, 2019.

Raffel C., Shazeer N., Roberts A., Lee K., Narang S., Matena M., Zhou Y., Li W., Liu P. J., “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”, 2020.

Rajapakse T. C., “Simple Transformers”, , https://github.com/ThilinaRajapakse/simpletransformers, 2019.

Robinson S. D., “Multi-Label Classification of Contributing Causal Factors in Self-Reported Safety Narratives”, *Safety*, 2018.

Shi D., Guan J., Zurada J., Manikas A., “A Data-Mining Approach to Identification of Risk Factors in Safety Management Systems”, *Journal of Management Information Systems*, vol. 34, p. 1054 - 1081, 2017.

Tanguy L., Tulechki N., Urieli A., Hermann E., Raynal C., “Natural language processing for aviation safety reports: From classification to interactive analysis”, *Computers in Industry*, vol. 78, p. 80-95, 2016. Natural Language Processing and Text Analytics in Industry.

Tulechki N., Natural language processing of incident and accident reports : application to risk management in civil aviation, phdthesis, Université Toulouse le Mirail - Toulouse II, September, 2015.

Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser L., Polosukhin I., “Attention Is All You Need”, 2017.

Wang A., Pruksachatkun Y., Nangia N., Singh A., Michael J., Hill F., Levy O., Bowman S. R., “SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems”, 2020.

Wang A., Singh A., Michael J., Hill F., Levy O., Bowman S. R., “GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding”, 2019.

Wolf T., Debut L., Sanh V., Chaumond J., Delangue C., Moi A., Cistac P., Rault T., Louf R., Funtowicz M., Davison J., Shleifer S., von Platen P., Ma C., Jernite Y., Plu J., Xu C., Scao T. L., Gugger S., Drame M., Lhoest Q., Rush A. M., “Transformers: State-of-the-Art

Natural Language Processing", *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, Online, p. 38-45, October, 2020.

Zhang X., Mahadevan S., "Ensemble machine learning models for aviation incident risk prediction", *Decision Support Systems*, vol. 116, p. 48-63, 2019.

APPENDIX: 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."

Examples of related metadata (mixture of analyst produced metadata and reporter produced metadata)

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