ENABLING NOISY LABEL USAGE FOR OUT-OF-AIRSPACE DATA IN READ-BACK ERROR DETECTION

Lakshmi Rajendram Bashyam, Alexander Blatt, Dietrich Klakow

Saarland University, Saarland Informatics Campus, Germany

ABSTRACT

Developing language understanding (NLU) methods for low resource domains is an ongoing challenge. The air-traffic control (ATC) domain is a paragon of this. There is a high pressure for automatized solutions to ease the workload of air-traffic controllers (ATCOs), but a low availability of open-source datasets. The available datasets contain mostly unlabeled transcripts, targeting automatic speech recognition (ASR) and cover just one or a few airspaces. Models trained on these airspaces might fail on unseen target airspace. We evaluate different methods to overcome this problem on the task of read-back error detection (RED), which uncovers mistakes in ATCO-pilot communication to prevent incidents. We generate noisy labels for our two stage RED approach, that combines data augmentation and noisy labels. This allows the use of unlabeled data of non-target airspaces to increase the performance on the target airspaces with a relative improvement of 35% over the baseline method.

***Index Terms*—** data augmentation, noisy labels, read-back-error detection, air-traffic control, class imbalance

1. INTRODUCTION

There is a high demand for machine learning (ML) based solutions in air traffic control to improve security, reliability, and safety. Degas et al. [1] provide an overview of the research in this area. To ensure high-quality machine learning tools, the European Union Aviation Safety Agency (EASA) published a guide for machine learning applications [2]. Assistant tools like autopilot or arrival manager [3] are already common tools to ease the daily work of pilots and air-traffic controllers. The high workload of air-traffic controllers (ATCOs) can lead to errors in communication and these errors can lead to incidents and accidents [4]. With air-traffic control (ATC) being responsible for 6-10% of aircraft crashes [5], incidents not included, assistant tools can play an important role in crash avoidance. Research projects like Malorca[1] or ATCO2[2] are focusing on developing such assistant tools and also databases to train them on [6]. Promising approaches rely on speech processing of ATCO and pilot communication. Outcomes of ATCO2 are for example a pipeline for collecting and annotating air-traffic communication [7] and a tool for recognizing callsigns in noisy air-traffic transcripts by using surveillance information [8]. These tools can reduce the ATCO workload and therefore indirectly reduce the chances of accidents.

A more straightforward approach is to employ automatic read-back error detection (RED) systems. The idea behind such a system is to directly detect mistakes in ATCO-pilot communication. A standard procedure of ATC communication involves the pilot reading back the command the ATCO has given. A pilot could for example answer with `Turning right 20 degrees` to the ATCO command `LUF674F turn right 20 degrees`. Each ATCO utterance should ideally start with the callsign of the addressed plane, in the example, this would be `LUF674F`. The callsign is followed up by a command `turn right` and the associated value `20 degrees`. The read-back of the command and value by the pilot is crucial, since it ensures, that there are no misunderstandings. To rule out, that the wrong pilot follows a command, the callsign of the plane is also read back in most of these cases.

In longer conversations, the read-back can also miss the callsign or include abbreviated versions of the callsign as stated by Blatt et al. [8], which complicates read-back error classification. To further complicate the matter, read-back errors occur just in 1-4% of the uttered commands [9–11]. Additionally there exist no publicly available datasets for RED. Furthermore, automatic speech recognition (ASR) datasets for ATC that could be labeled, might not contain data from desired the target airspaces. All this makes it difficult to train machine-learning based methods for read-back error systems. This is one of the reasons, why other ML based systems focus on binary read-back error classification [11–15].

In this paper, we investigate methods to handle this low-resource problem and propose to the best of our knowledge the first benchmarks for a fully ML-based multi-class read-back error recognition system.

2. RELATED WORK

Because of the severe consequences, causes of ATC errors are the target of several studies. Marrow et al. [16] identify amongst others the length of an ATC message and the amount of traffic as causes for errors in ATC communications. Cardosi et al. [4] uncover wrong pilot expectations, pilots sharing the same frequency

[1]MALORCA Homepage: `https://www.malorca-project.de/`

[2]ATCO2 Homepage: `https://www.atco2.org/`

979-8-3503-0689-7/23/$31.00 ©2023 IEEE

and a high controller workload as additional factors. In a more recent work by Wu et al. [17] a correlation between pilot accents and miscommunication is stated.

An early machine learning based read-back error detection method is implemented by Chen et al. [12]. They propose an automatic speech recognition (ASR) based system, that features a GUI to display read-back error alerts. Jia et al. [14] use an LSTM-based model for binary read-back error detection of transcribed Chinese ATC utterances. They achieve an accuracy of 94%. However, due to the seldom occurrence of read-back errors, the system is evaluated on synthetically generated read-back error samples. A two-step approach is taken by Helmke at al. [15]. In the first step, the ATC transcripts are converted either rule-based or transformer [18] based into a standardized ATC phraseology. They use a rule-based model for identifying individual use cases which also include read-back error cases. Alternatively, a BERT [19] based approach is used for read-back error detection. But due to the low occurrence of read-back errors and the resulting class imbalances in the training data, they opt for binary classification in their machine-learning-based approach. On real-life ops-room recordings, they reach an F1 score of 47% when combining the data-driven and rule-based read-back error detection system.

In contrast to previous works, our system relies purely on a machine-learning based approach. By employing techniques to handle class imbalance and using out-of-airspace data, our system is able to effectively detect different read-back error classes. The classes used in our RED are the result of grouping operational scenarios with different degrees of severity by our ATC experts. They provide the ATCO with more feedback than a binary RED.

One of those techniques is generating noisy labels. We especially build on two previous noisy label works. Firstly, Zhu at al. [20] have shown that noisy labels can be used with BERT without using advanced noise handling methods, like noise matrices. Secondly Goh et al. [21] used a two-step approach, in which they fine-tune their model in a first step on noisy labeled data and then fine-tune it a second time on clean data, to avoid overfitting on the noisy labels.

3. METHODS

In the following, we will describe how we build our dataset and describe the methods used for read-back error detection.

3.1. Read-back Error Classes

In the scope of this paper, we focus on pair-wise read-back error detection, meaning, that we look at errors occurring in an answer from a pilot to an ATCO command. We consider 5 different classes for read-back error detection. Examples of ATCO-pilot utterance pairs for each class are given in Table 1.

If no read-back errors are detected, the utterance pair is labeled as `Correct`. If there are two commands given by an ATCO and just one is correctly read back, this is `Partial` read-back. A pair is labeled as `Wrong` if a pilot reads an incorrect command back, for example, the wrong turning angle. If there is no read-back at all, it is labeled as `Missing`. `Wrong Pair` covers two possible cases. In one case, the pilot utterance is completely unrelated to the ATCO command, this for example happens, when a new plane enters the airspace and the pilot makes contact with the ATC just after a command is spoken to another plane. The second, more problematic case is, that the wrong pilot answers a command which was not meant for him. The analysis of our dataset has shown, that this is the case for less then 10% of the `Wrong pair` samples.

Table 1: Read-back Error Classes

Error Class	Example
Correct	ATCO: AFR617 contact Maastricht 132.755 bye bye PILOT: 132 755
Partial	ATCO: 7AW climb flight level 300 and turn right by 10 degrees PILOT: Turning right 10 degrees
Wrong	ATCO: Beauty 4306 descend to flight level 250 PILOT: Descend flight level 350 confirm
Missing	ATCO: Roger, call you back very shortly maintain 330 PILOT: thank you
Wrong Pair	ATCO: KLM9F climb flight level 310 PILOT: did you just call DLH89F

3.2. Data Labeling

The ATC transcripts for building our corpus are collected from two ATC corpora, namely the LiveATC and the LDC-atcc corpus. The LDC-atcc corpus [22] consists of ATC communication and transcripts from the airspaces surrounding the following airports: Dallas Fort Worth International (KDFW), Logan International (KBOS) and Washington National Airport (KDCA). The LiveATC dataset, collected during the ATCO2 project [7], consists of transcripts from the ATC radio, recorded from the LiveATC website [3]. LiveATC provides live streams of ATC communications for different airport airspaces. For the read-back error detection dataset, samples from Amsterdam Airport Schiphol (EHAM), Dublin Airport (EIDW), Göteborg Landvetter Airport (ESGG), Zurich Airport (LSZH), and Stockholm Västerås Airport (ESOW) are used.

Both datasets are pooled and ATCO-pilot pairs are extracted based on timestamps. To label the pairs efficiently, a dataset of

[3]LiveATC website: `https://www.liveatc.net/`

Table 2: Class distribution of samples in the initial pool, collected by active learning (AL), data augmentation (Aug), and rule based system (Noisy)

Method	Partial	Missing	Correct	Wrong	Wrong pair	Total
Initial	93	66	712	41	40	952
AL	84	58	138	6	31	317
Aug	763	499	0	514	0	1776
Noisy	1188	1143	4469	1898	1407	10105

952 samples is build by manually categorizing these pairs into the classes listed in Table 1. The manual labelling is performed by the author and supported by experts of the ATC domain to ensure proper labeling and the selection of proper read-back error classes. The rest of the samples is labeled using active learning (AL). We first train a bert-base-uncased model on the initial data pool and then use the prediction entropy technique [23] to select 20 additional samples out of the unlabeled pool. This cycle is reiterated, with the training pool growing with each iteration, until a total of 317 extra samples are acquired. This raises the likelihood of discovering informative sample pairs within the data pool.

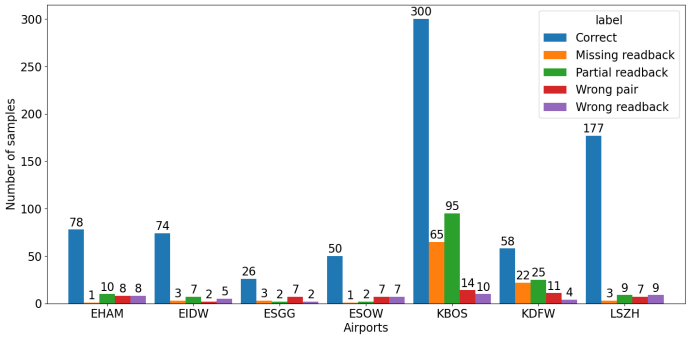
After active learning, the sample pool consists of 1232 samples as Table 2 shows. It should be mentioned, that the AL does not change the label distribution significantly.

The label distribution for the different airports after the active learning step is displayed in Figure 1. As already discussed in previous works, the distribution is unbalanced, with the `Correct class` making up more than 60% of the samples. One way to address this is to perform data augmentation as described in subsection 3.4

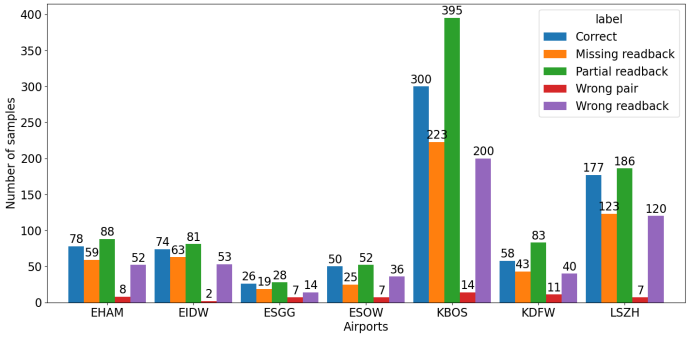
3.3. Number Standardization

A closer look at Table 1 shows that the comparison of command values between ATCO and pilot transcripts can be sufficient to classify the pair. For the `Correct` read-back in Table 1 for example, the pilot and command utterance contain the same value, `132755`. For the distinction between the `Wrong Pair` class and the other classes, the comparison of the callsigns in ATCO and pilot transcripts is equally important. Since command values and callsigns contain both digits, a classification could be further simplified by just matching digits between the pilot and ATCO utterance.

The main problem with this concept is, that numbers are not spelled out in a standardized format. The number 444 could be uttered for example as `four four four`, `four hundred four` or `triple four`. This makes a matching difficult. To overcome this issue, we format each number in a standard format by splitting it into its individual digits by using a tool provided by Brandhsu et al. [24].



(a) Label distribution across the different airport airspaces.



(b) Label distribution over airport airspaces after data augmentation.

Fig. 1: Label distribution before (a) and after (b) data augmentation.

3.4. Data Augmentation

To address the low occurrence of read-back error cases, we augmented the read-back error classes in the training data. No augmentation is done for the test data, to ensure a realistic testing scenario. For the `Wrong` and `Missing` class, we formulate search patterns for the commands and the corresponding values, similar to regular expressions. For `Wrong` read-back, the values in the pilot read-back of the `Correct` pairs are altered by changing numbers via substituting, deleting or adding digits, e.g `turn 10 degrees` is changed to `turn 20 degrees`. For `Missing` read-back, the command and value in the `Correct` read-back pairs are completely removed. To augment `Partial` read-back, ATCO-pilot pairs are generated by combining a callsign with two of the isolated commands and values to create an ATCO transcript. For the pilot read-back, just one of the issued commands is used. The `Correct` and `Wrong pair` labels are not augmented, since the read-back error classifier already works sufficiently well on these classes before augmentation.

Figure 1(b) shows the label distribution of each airport airspace after the augmentation. In comparison with Figure 1(a), the higher frequency of read-back error cases is clearly visible. This leads to a more balanced training data set, which prevents overfitting on the `Correct` class.

3.5. Noisy Labeling

Transformer-based models, require a sufficient amount of training data to achieve competitive results, especially for an unseen domain. Annotating error classes for RED on the other hand, requires a significant amount of effort and needs experts from the air traffic control field, since ATCO utterances can contain multiple commands. It is crucial to carefully verify the presence of all these commands in order to identify the type of error correctly. An alternative to this time consuming labeling are noisy labels. For our noisy labeling approach, we collect the unlabelled ATCO-pilot samples from the LDC-atcc corpus, namely from the Logan and DFW airspaces in the United States. Our rule-based system for generating noisy labels consists of the following steps:

1. First, we extract the command values from the ATCO commands for number groups and special word groups. For example, the command `bizex three twenty nine turn left heading one correction zero niner zero` would have `left`, and `zero niner zero` as extracted groups. The script is carefully constructed to cover all possible ATCO commands.

2. In the next step, we match the extracted ATCO command values with the pilot read-back. If all the extracted command values are present in the read-back in the same order, it is classified as a `Correct` read-back. If none of the command values are present, it is a `Missing` read-back, and if only a fraction of several commands is read back by the pilot, it is a `Partial` read-back. If the order of command values is shuffled or if one or more numbers/words are missing, it is classified as a `Wrong` read-back.

3. To identify `Wrong pair` samples, we extract the callsign from the ATCO command, including the aircraft name and code and match it with the pilot read-back. This helps to recognize if the pilots read-back contains complete, partial or missing callsigns.

4. `Wrong pair` read-back classes occur when a different pilot than expected responds to the ATCO command (see Table 1) or when a new pilot starts communicating on a specific frequency. The missing callsign along with missing command values identified in the previous steps are an indicator for a different pilot responding to the ATCO command. A greeting in a pilot read-back indicates that a new pilot is speaking, since a greeting never happens after an ATCO command is given.

For reproducibility, our noisy labeling method is made publicly available [4]. In the next section we will explain how we use the noisy labeled data and our rule-based method for read-back error detection.

[4] https://github.com/uds-lsv/RulebasedRED

4. EXPERIMENTAL SETUP

We are investigating the scenario, where the read-back error system is only tested on unseen airspaces, to examine the inter-airspace transferabilty of the different algorithms. We do not generate augmented data for the test airspace to ensure a realistic test scenario. Figure 1 shows, that without augmentation, there exist just a few samples for the majority of the read-back error classes per airport. For each test airport, we use the (augmented) data for training data whereas the validation data only consists of high quality manually annotated data. Both train and validation data do not contain any data from the test airspace nor is it used to perform augmentation. In the cross-validated experiments, we take the mean of all airports for three seeds and present the mean and standard deviation of it.

BERT (bert-base-uncased) is used as read-back error classifier, similar to Helmke et al. [15]. The transcript pairs are fed into the recognizer in the following format: `[CLS] ATCO transcript [SEP] Pilot transcript [SEP]`. The recognizer is trained with the ADAM optimizer with a learning rate of $2e^{-5}$ and cross-entropy loss with early stopping is used to avoid overfitting.

We test six different methods to improve the RED. The first method is our rule-based system to create the noisy labels, called "Rule-based" in Table 3, which is directly applied to the test data. The second method is a BERT-baseline without any methods applied to handle the class imbalance, respectively the low-resource scenario. In the third method, weighted cross entropy loss (w. CE) is used to handle the class imbalance. The fourth method consists of augmenting the training data as described in subsection 3.4. The fifth method uses the noisy labeled data, described in subsection 3.5. Zhu et al. [20] have shown that special noise-handling methods like Co-teaching or noise matrices are not needed for BERT models and can even harm the performance. However, if there is clean data available for training, Goh et al. [21] have shown that using a two-stage training process can increase the performance, if the model is finetuned first on the noisy labels and then on clean data. This is due to the reason, that in most cases there is more noisy than clean labeled data. Experiments have shown that models will overfit on the noisy labels, which degrades the system's performance. We therefore apply the two-step approach by Goh et al. [21] to avoid overfitting.

In the sixth method, we make use of all the available datasets, including manually annotated, augmented and noisy data, in an two-stage noisy + augmented training. In this approach, the noisy labels are used to initially fine-tune a pre-trained BERT-base-uncased model, which is then further fine-tuned with both augmented and manually annotated datasets. It should be noted, that the augmented data is only included in the training, while the validation set consists solely of manually annotated data. The precision, recall, F1-score and accuracy metric for each of the model used in the experiment is calculated.

Table 3: Scores for training without target airport for the different data handling methods. Scores are given as the macro average of all read-back error classes. The experiments are repeated thrice and the mean is given. The standard deviation is given in brackets.

Method	Precision	Recall	F1	Accuracy
Rule-based	38.46	45.26	38.46	61.21
Baseline	49.94 (±0.8)	44.05 (±0.9)	43.77 (±1.3)	73.6 (±0.3)
w. CE	45.8 (±3.6)	45.11 (±4.9)	42.97 (±3.9)	74.23 (±0.5)
Aug	49.5 (±0.2)	51.6 (±2.3)	45.28 (±1.1)	63.03 (±4.1)
Two-stage noisy	54.68 (±2.7)	49.73 (±0.8)	49.35 (±0.7)	**75.90 (±0.2)**
Two-stage noisy + aug	**60.8 (±1.7)**	**66.98 (±2.5)**	**59.11 (±1.8)**	73.98 (±7.1)

5. RESULTS

We show in the following the results of training our RED system with the six different methods explained in section 4 to evaluate the inter-airspace transferability of the methods. Table 3 shows the precision, recall, F1 scores and accuracies for each method.

The results show, that our rule-based system for producing noisy labels performs reasonably well with a 5% lower F1 score than the baseline model. The best F1, precision and recall scores are reached for the two-stage approach with noisy labels and augmented data. This method outperforms the baseline with over 15%. Even the second best algorithm, the two-stage noisy approach, has still a 10% lower F1 score than the best method. This is surprising since just using augmented data gives less than 2% improvement over the baseline. These findings underline the importance of combining the augmentation approach with the two-step noisy approach.

To get a better understanding of the class-wise performance, the F1, recall and precision scores for each class are shown in Table 4.

Looking at the precision scores, there is no model that clearly outperforms the others, but the rule-based labeling approach performs best for two classes, namely `Partial` and `Wrong`. This indicates, that the rules for those classes are well designed, since they filter out the other classes effectively. This holds true especially for the `Wrong` class, where just the two-stage noisy + augmented approach reaches a similar precision value. But the main goal of RED is incident avoidance. When it comes to incident avoidance, the recall values are more important than the precision values, since a high recall value ensures, that no error case is missed. For all the error cases, except for `Wrong pair`, the two-stage approach with noisy labels and augmented data shows the highest scores. For the `Correct` and `Wrong pair` class the weighted cross-entropy reaches the highest score. The same pattern can be seen for the F1 scores. It should be however noted, that the two-stage approach with noisy labels and augmented data outperforms all other methods on the `Wrong` class by a considerable margin, probably benefiting from the rule-based noisy labels. Interestingly, the pure two-stage noisy labels approach cannot reach similar performance levels, probably due to the small number of clean labels. To put the performance of our best method into perspective, we compare it with the RED systems presented by Helmke et al. [15]. Their solely machine-learning based system in [15] reaches for comparison on a dataset consisting of Isavia ops-room transcripts, which even includes transcripts of the target airport, an F1 score of 29% for the binary classification of `read-back OK` and `read-back ERROR`. Their highest scoring hybrid system, which combines a rule-based and ML approach, reaches an F1 score of 47%. Our two-stage approach with noisy labels and augmented data reaches without ever having seen the target airport an F1 score of 59.11% with a low standard deviation of 1.8%, but for multi-class read-back error detection, instead for binary detection.

To understand better why the two-stage approach with noisy labels and augmented data performs so well, the F1 scores for all methods are plotted for each airport airspace in Figure 2. In the figure, the difference between the American (KBOS, KDFW) and the European (EHAM, EIDW, ESGG, LSZH, ESOW) airspaces is clearly visible. For the American airports, the performance difference between the different methods is not as big as for the European airports, but it should be mentioned, that the baseline methods performs already quite well on KBOS and KDFW, indicating, that the American corpora are less complex. This could also be the result of the higher number of samples for read-back error classes for the American airports compared to the European airports as seen in the label distribution Figure 1.

But the more important observation is, that the F1 scores on the European airports drastically improve when using the two-stage approach with noisy labels and augmented data. Interestingly, just for EHAM, the two-step noisy approach reaches the same performance as the two-stage approach with noisy la-

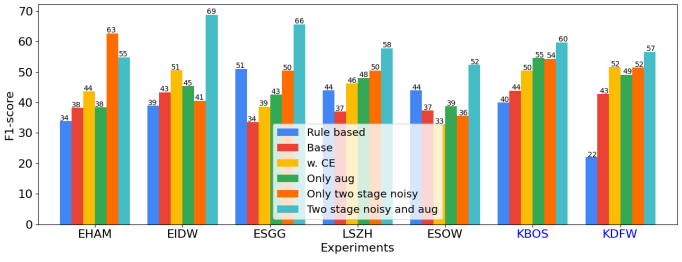


Fig. 2: F1 scores for the individual airport airspaces. The American airports are marked in blue

Table 4: Mean F1, recall and precision scores over all airports for the different data handling methods with scores for each read-back error class. The experiments are repeated thrice and the mean is given. The standard deviation is given in brackets.

	Method	Correct (%)	Partial (%)	Wrong pair (%)	Missing (%)	Wrong (%)
Precision	Rule-based	86.14	**66.85**	27.01	15.82	**32.74**
	Baseline	78.77 (±0.42)	33.90 (±5.5)	81.39 (±0.6)	**52.85 (±3.9)**	2.77 (±2.0)
	w. CE	81.08 (±0.7)	18.7 (±4.1)	79.9 (±11.0)	40.21 (±19)	9.3 (±6.2)
	Aug	82.3 (±1.8)	27.17 (±4.4)	**84.33 (±2.2)**	41.44 (±4.9)	12.30 (±0.4)
	Two-stage noisy	82.72 (±1.1)	57.49 (±0.8)	74 (±1.4)	49.7 (±0.6)	20.69 (±11.1)
	Two-stage noisy + Aug	**88.13 (±0.5)**	57.49 (±0.8)	82.94 (±0.9)	44.47 (±5.8)	31.24 (±1.8)
Recall	Rule-based	66.85	52.89	27.01	15.82	54.86
	Baseline	87.2 (±0.8)	36.45 (±7.7)	53.6 (±7.5)	41.72 (±3.8)	1.2 (±0.8)
	w. CE	**87.58 (±2.3)**	24.10 (±1.5)	**64.54 (±2.1)**	38.46 (±18.8)	10.09 (±8.1)
	Aug	67.16 (±5.8)	31.75 (±3,4)	56.08 (±3.9)	63.41 (±6.9)	39.93 (±5)
	Two-stage noisy	85.0 (±0.9)	50.6 (±5.3)	60.4 (±3.9)	46.9 (±2.5)	5.77 (±1.7)
	Two-stage noisy + Aug	76.49 (±0.7)	**62.25 (±3.2)**	57.49 (±0.9)	**78.24 (±8.1)**	**60.44 (±6.2)**
F1	Rule-based	73.22	40.17	25.3	17.11	36.51
	Baseline	82.48 (±0.61)	33.95 (±6.3)	57.3 (±5.7)	43.48 (±1.3)	1.65 (±0.1)
	w. CE	**83.5 (±8.0)**	20.46 (±2.9)	**66.33 (±4.6)**	35.77 (±16.6)	8.73 (±7.1)
	Aug	72.75 (±4.1)	27.57 (±1.6)	63.31 (±3.5)	45.11 (±1.8)	17.65 (±0.2)
	Two-stage noisy	82.98 (±0.4)	47.53 (±4.2)	61.4 (±2.1)	46.79 (±1.7)	7.77 (±3.2)
	Two-stage noisy + Aug	80.86 (±0.9)	**58.77 (±0.3)**	63.8 (±0.4)	**52.3 (±5.7)**	**39.69 (±3.1)**

bels and augmented data. This indicates, that there is an influence of the domain or air-space mismatch, between the European airspaces and the noisy labels, obtained from the American airspaces. By using the augmented data in the second step of the two-stage approach with noisy labels, the American bias, that is introduced by the noisy labels is cured.

6. CONCLUSION

In this work, we demonstrate the first fully machine learning based model for multi-class read-back error detection. In contrast to previous works who propose machine learning based models for binary read-back error classification, our model is capable of distinguishing the classes `Correct, Partial, Wrong, Missing` read-back and `Wrong Pair`. We evaluate different methods to overcome, the highly unbalanced and low-resource scenario for the read-back error classes. We introduce a class-wise data augmentation method and a rule-based noisy labeling approach to generate noisy labeled data. We incorporate this data in our two-step training approach using noisy labels in the first step and augmented data in the second step. We show, that this method reaches an F1 score of 59.11% on unseen airspaces and outperforms the other investigated methods, like the two-step noisy label training without augmented data, by at least 10%. Furthermore, we show that this method performs consistently well over all error classes, while the other methods show performance drops, especially for the `Wrong` read-back class. Additionally, we can show, that using augmented data in the second step of the two-step training is crucial for out-of-airspace noisy labeled data, since it allows to overcome the bias of the airspace-mismatch. Therefore our proposed two-stage method with noisy labels and augmented data is an effective way to improve read-back error detection, even in low-resource scenarios. We additionally want to emphasize, that this method is not restricted to read-back error detection and could be also used in other low-resource domains, where there is a domain mismatch between noisy labels and the test data.

7. FUTURE WORK

Initial experiments with other evaluation metrics for imbalanced datasets, like Focal loss [25] did not show significant improvements over weighted cross-entropy loss, but we will explore additional metrics in future experiments. We also want to address the low scores of `Wrong` read-back by improving our data augmentation method. To reduce the occurrence of false alarms for read-back errors, an additional focus lies on improving the accuracy scores, without compromising on the F1 scores. This is equivalent to reaching higher F1 scores for the majority class `Correct`, which covers over 90% of the samples in a real-life ATC communication. We additionally want to apply our two-stage method with noisy labels and augmented data to other low-resource domains and evaluate it against pure data augmentation and pure noisy label training.

8. REFERENCES

[1] Augustin Degas, Mir Riyanul Islam, Christophe Hurter, Shaibal Barua, Hamidur Rahman, Minesh Poudel, Daniele Ruscio, Mobyen Uddin Ahmed, Shahina Begum, Md Aquif Rahman, Stefano Bonelli, Giulia Cartocci, Gianluca Di Flumeri, Gianluca Borghini, Fabio Babiloni, and Pietro Aricó, "A Survey on Artificial Intelligence (AI) and eXplainable AI in Air Traffic Management: Current Trends and Development with Future Research Trajectory," *Applied Sciences (Switzerland)*, vol. 12, no. 3, pp. 1295, jan 2022.

[2] European Union Aviation Safety Agency, "EASA Concept Paper: First usable guidance for Level 1 machine learning applications," , no. 1, pp. 1–174, 2021.

[3] Oliver Ohneiser, Hartmut Helmke, Heiko Ehr, Hejar Gürlük, Michael Hössl, and Thorsten Mühlhausen, "Air Traffic Controller Supportby Speech Recognition," *Advances in Human Aspects of Transportation: Part II*, vol. 16, no. July, 2021.

[4] Kim Cardosi, Paul Falzarano, and Sherwin Han, "Pilot-controller communication errors: An analysis of Aviation Safety Reporting System (ASRS) reports," 1998.

[5] Lejla Nikšić and Ebru Arıkan Öztürk, "U.S./Europe Comparison of Atc-Related Accidents and Incidents," *International Journal for Traffic and Transport Engineering*, vol. 12, no. 2, pp. 155–169, apr 2022.

[6] Juan Zuluaga-Gomez, Karel Veselý, Igor Szöke, Petr Motlicek, Martin Kocour, Mickael Rigault, Khalid Choukri, Amrutha Prasad, Saeed Sarfjoo, Iuliia Nigmatulina, Claudia Cevenini, Pavel Kolčárek, Allan Tart, and Jaň Cernocký, "ATCO2 corpus A Large-Scale Dataset for Research on Automatic Speech Recognition and Natural Language Understanding of Air Traffic Control Communications," nov 2022.

[7] Martin Kocour, Karel Veselý, Igor Szöke, Santosh Kesiraju, Juan Zuluaga-Gomez, Alexander Blatt, Amrutha Prasad, Iuliia Nigmatulina, Petr Motlíček, Dietrich Klakow, Allan Tart, Hicham Atassi, Pavel Kolčárek, Jan Černocký, Claudia Cevenini, Khalid Choukri, Mickael Rigault, Fabian Landis, Saeed Sarfjoo, and Chloe Salamin, "Automatic Processing Pipeline for Collecting and Annotating Air-Traffic Voice Communication Data," *Engineering Proceedings*, vol. 2, no. 1, pp. 8, dec 2022.

[8] Alexander Blatt, Martin Kocour, Karel Veselý, Igor Szöke, and Dietrich Klakow, "Call-Sign Recognition and Understanding for Noisy Air-Traffic Transcripts Using Surveillance Information," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2022-May, no. 864702, pp. 8357–8361, 2022.

[9] M. Cardosis, "An Analysis of Tower (Local) Controller - Pilot Voice Communications," , no. June, 1994.

[10] O Veronika Prinzo, Alfred M Hendrix, and Ruby Hendrix, "The Outcome of ATC Message Length and Complexity on En Route Pilot Readback Performance," 2009.

[11] Hartmut Helmke, Matthias Kleinert, Shruthi Shetty, Oliver Ohneiser, Heiko Ehr, Hörur Arilíusson, Teodor S Simiganoschi, Amrutha Prasad, Petr Motlicek, Karel Veselý, Karel Ondrej, Pavel Smrz, Julia Harfmann, and Christian Windisch, "Readback Error Detection by Automatic Speech Recognition to Increase ATM Safety," in *14th USA/Europe Air Traffic Management Research and Development Seminar, ATM 2021*, 2021.

[12] Shuo Chen, Hunter Kopald, Ronald S. Chong, Yuan Jun Wei, and Zachary Levonian, "Read back error detection using automatic speech recognition," *12th USA/Europe Air Traffic Management R and D Seminar*, 2017.

[13] Fangyuan Cheng, Guimin Jia, Jinfeng Yang, and Dan Li, "Readback error classification of radiotelephony communication based on convolutional neural network," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 10996 LNCS, pp. 580–588.

[14] Guimin JIA, Fangyuan CHENG, Jinfeng YANG, and Dan LI, "Intelligent checking model of Chinese radiotelephony read-backs in civil aviation air traffic control," *Chinese Journal of Aeronautics*, vol. 31, no. 12, pp. 2280–2289, dec 2018.

[15] Hartmut Helmke, Karel Ondřej, Shruthi Shetty, Hörur Arilíusson, Teodor S Simiganoschi, Matthias Kleinert, Oliver Ohneiser, Heiko Ehr, Juan-Pablo Zuluaga, and Pavel Smrz, "Readback Error Detection by Automatic Speech Recognition and Understanding Results of HAAWAII project for Isavia's Enroute Airspace," Tech. Rep., 2022.

[16] Daniel Morrow, Alfred Lee, and Michelle Rodvold, "Analysis of Problems in Routine Controller-Pilot Communication," *The International Journal of Aviation Psychology*, vol. 3, no. 4, pp. 285–302, 1993.

[17] Qiong Wu, Brett R.C. Molesworth, and Dominique Estival, "An Investigation into the Factors that Affect Miscommunication between Pilots and Air Traffic Controllers in Commercial Aviation," *International Journal of Aerospace Psychology*, vol. 29, no. 1-2, pp. 53–63, apr 2019.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*. 2017, vol. 2017-Decem, pp. 5999–6009, arXiv.

[19] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*. 2019, vol. 1, pp. 4171–4186, arXiv.

[20] Dawei Zhu, Michael A Hedderich, Fangzhou Zhai, David Ifeoluwa Adelani, and Dietrich Klakow, "Is BERT Robust to Label Noise? A Study on Learning with Noisy Labels in Text Classification," in *Insights 2022 - 3rd Workshop on Insights from Negative Results in NLP, Proceedings of the Workshop*, 2022, pp. 62–67.

[21] Garrett B. Goh, Charles Siegel, Abhinav Vishnu, and Nathan Hodas, "Using rule-based labels for weak supervised learning: A chemnet for transferable chemical property prediction," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, New York, NY, USA, 2018, KDD '18, p. 302–310, Association for Computing Machinery.

[22] John J. Godfrey, "Air Traffic Control Complete," 1994.

[23] Alex Holub, Pietro Perona, and Michael C Burl, "Entropy-based active learning for object recognition," in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops*, 2008.

[24] Brandhsu, "A simple, deterministic, and extensible approach to inverse text normalization for numbers," https://github.com/barseghyanartur/itnpy, 2022.

[25] Akhilesh Gupta, Nesime Tatbul, Ryan Marcus, Shengtian Zhou, Insup Lee, and Justin Gottschlich, "Class-Weighted Evaluation Metrics for Imbalanced Data Classification," 2020.