Analyzing and overcoming low-resource and domain specific problems in the air-traffic control speech processing pipeline

Dissertation zur Erlangung des Grades

des Doktors der Ingenieurwissenschaften (Dr.-Ing.)

der Naturwissenschaftlich Technischen Fakultät

der Universität des Saarlandes

von Alexander Blatt

Saarbrücken, April 15, 2025

TAG DES KOLLOQUIUMS: 14.10.2025

DEKAN: Prof. Dr.-Ing. Dirk Bähre

BERICHTERSTATTER: Prof. Dr. Dietrich Klakow

Prof. Dr.-Ing. Georg Frey

AKAD. MITARBEITER: Dr.-Ing. László Levente Tóth

VORSITZ: Prof. Dr. Andreas Schütze

ORT:

Saarbrücken

Abstract

Roughly a decade ago, machine learning-based (ML) assistance solutions for air-traffic control (ATC) became a research focus. Since then, many publications aim to reduce the workload of air-traffic controllers (ATCOs). Especially works targeting automatic speech recognition (ASR) and natural language processing have shown drastic improvement in recent years. However, most of the research is focused on the improvement on ATC benchmark datasets and not on key requirements for real-world ML ATC systems such as robustness, explainability and privacy. Addressing these mitigates the risks of incidents and ensures that models are aligned with data protection laws.

This thesis therefore focuses on addressing these requirements within an ATC speech processing pipeline. Starting at the beginning of the pipeline, we investigate the influence of acoustic and lexical differences between ATC datasets on ATC-ASR models. Going further in the pipeline, we compare the robustness of combined ASR and speaker role detection architectures. At the end of the pipeline, we propose robust call-sign recognition methods and show how to train a read-back error detection system that generalizes well to unseen airspaces. Finally, we demonstrate at the example of ACTO stress detection that implementing privacy measures in the pipeline does not hurt its performance. The new insights, training procedures and architectures of this thesis bring ML based ATC support systems closer to operation.

Zusammenfassung

Vor einem Jahrzehnt rückten maschinell lernende (ML) Assistenzlösungen für die Flugverkehrskontrolle (ATC) in den Forschungsfokus. Seitdem zielen Veröffentlichungen darauf ab, die Belastung von Fluglotsen (ATCOs) zu reduzieren. Besonders bei der automatischen Spracherkennung (ASR) und der Verarbeitung natürlicher Sprache gab es zuletzt drastische Verbesserungen. Jedoch liegt hier der Fokus auf der Verbesserung auf ATC-Benchmark-Datensätzen und nicht auf Schlüsseleigenschaften von ML-ATC-Systemen wie Robustheit, Erklärbarkeit und Datenschutz. Die Berücksichtigung dieser Aspekte vermindert Zwischenfällen und harmonisiert Modelle mit den Datenschutzgesetzen.

Diese Arbeit adressiert diese Eigenschaften innerhalb einer ATC-Sprachverarbeitungspipeline. Am Anfang der Pipeline untersuchen wir den Einfluss akustischer und lexikalischer Unterschiede zwischen ATC-Datensätzen auf ATC-ASR-Modelle. Anschließend vergleichen wir die Robustheit kombinierter ASR- und Sprecherrollenerkennungsarchitekturen. Am Ende der Pipeline schlagen wir robuste Methoden zur Rufzeichenerkennung vor und zeigen, wie man ein Read-back-Fehlerdetektionssystem trainiert, das auf unbekannte Lufträume generalisiert. Schließlich demonstrieren wir am Beispiel der ACTO-Stresserkennung, dass Datenschutzmaßnahmen nicht die Leistung der Pipeline beeinträchtigen. Die neuen Erkenntnisse, Trainingsverfahren und Architekturen bringen ML-basierte ATC-Unterstützungssysteme näher an den operationellen Einsatz.

Science is a collaborative effort.
The combined results of several people
working together is often much more effective
than could be that of an individual scientist
working alone.

— **John Bardeen**

Acknowledgments

First, I want to thank my supervisor Dietrich Klakow, who gave me the opportunity to work in his research group. Dietrich allowed me to explore my own ideas, but also kept me on track, whenever I went to far of course. When I needed supervision, he was there to support me. Dietrich, also managed to gather a great team. A special thanks goes to my colleagues Marius Mosbach, Badr Abdullah, Volha Petukhova and Nicolas Louis, who where always there, when I needed advice, and in Nicos case help. I want to thank Aravind Krishnan for the great collaboration. A big thanks goes also to my other colleagues Dana Ruiter, Miaoran Zhang, Anupama Chingacham, Paloma Garcia de Herreros, Vagrant Gautam, Michael Hedderich, David Adelani, Jesujoba Alabi, Dawei Zhu, Florian Dietz and Julius Steuer. You all made my time at LSV very special and I will be always gratefully for that.

A thanks goes also to my collaborators from the ATCO2 project, Martin Kocour, Karel Veselý and Igor Szöke. I am also very gratefully for the great collaboration with Deutsche Flugsichering, here I want to give a special thanks to Konrad Hagemann. I also want to thank Rüdiger Ehrmanntraut and Guy Bormann from Eurocontrol Mastricht for the long collaboration and all their feedback and input. A special thanks goes also to Roland Tichy and Pieke Satijn for the great collaboration.

One of the greatest part to be a PhD student is the possibility to supervise students, I want to thank Lakshmi Bashyam, Janaki Viswanathan Sushmita Nair and Sunny Rahul for their enthusiasm,ideas and hard work. It was an honor to supervise you.

And finally I want to thank my family for the support. I want to thank my parents, who always supported my non-linear career path and my little brother to make me want to catch up with him. But my biggest thanks by far goes to my wife Marie-Louise, without her, I would not be able to write these words. Thank you for your unconditional love and support. And also thanks to my children Bruno and Ella who can brighten the greyest days.

A big thanks to all of you!

Contents

1

Introduction

Contents

1.1 Motivation

Artificial neural networks (ANN) are already an normal part of our lives, but took a long time from their invention to mature to a useful technology. The history of ANNs dates back to the first half of the last century. The neuron as logical element was already introduced in 1943 and modeled after a real neuron. It outputs either `true` or `false`, depending on the summation of the inputs reaching a threshold (McCulloch et al., 1990). The first implemented ANN, the perceptron followed a few years later in 1957 (Frank, 1958). Despite these early inventions, it took nearly 50 years until the development of ANN gained momentum. Although further important inventions were made in this period. In fact, John Hopfield and Geoffrey Hinton have been rewarded with the Nobel Prize in Physics in 2024 for setting foundations for machine learning, respectively neural networks. They used physics to train NNs, by describing the training process as a search for the state with

minimum energy (Hopfield, 1982; Ackley et al., 1985). Other important inventions are for example the long short-term memory (LSTM) network (Hochreiter et al., 1997) or the backpropagation algorithm, the basis for efficient ANN training (Rumelhart et al., 2019), but further development stalled due to the missing hardware. The introduction of more powerful graphics processing units (GPUs) in the new millennium and the compute unified device architecture (CUDA) in 2007, as parallel computing platform, allowed to accelerate the ANN training process, by parallelizing it. This enabled the development of bigger and more importantly deeper models, heralding the era of deep learning (Lecun et al., 2015).

The introduction of deep convolutional neural networks (CNNs) like AlexNet (Krizhevsky et al., 2012) drastically enhanced the image recognition capabilities of of deep neural networks (DNNs). An even bigger paradigm shift was caused by the introduction of transformers (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Aidan N Gomez, et al., 2017a), which led to a drastic increase in performance in natural language processing (NLP) (Devlin, M. W. Chang, et al., 2019). An even bigger impact from today's perspective was the improvement of language modeling (LM) (OpenAI et al., 2023), leading tho large language models (LLMs). Another task that benefited from transformers is automatic speech recognition (ASR) (Radford et al., 2022). The general performance improvement of DNNs in the late 2010s and early 2020s resulted in them surpassing humans in several task, from playing Go (Silver et al., 2017) to more practical tasks like image classification or English understanding (Perrault et al., 2024).

At least since the upcoming of ChatGPT, DNNs reached the general public (Ray, 2023). Simultaneously, they are also becoming increasingly important for multiple industry sectors (Jan et al., 2023). In the manufacturing industry, DNNs are the main accelerator for production optimization (De Simone et al., 2022). The automotive industry would not be able to sell autonomous driving cars without neural networks (Bathla et al., 2022). But probably the most impactful utilization of DNNs happened in the medical industry, where neural network based vaccine engineering ended the global Covid-19 pandemic (Sharma et al., 2022). Apart

from these popular applications, there are also lesser-known machine-learning applications. One field, where DNNs are already used is the air-traffic control (ATC) domain.

Air-traffic controllers (ATCOs) manage hundreds of flights every day. A typical en-route controller, which manages flights during the cruise phase, handles for example roughly 30 flights simultaneously (Request et al., 2006). An ATCO is therefore responsible for a save journey of thousands of people every day. The amount of ATC traffic was and will be continuously increasing for decades (Zhang et al., 2012). Since higher traffic correlates with higher mental workload (Corver et al., 2016), the performance of an ATCO decreases and the probability for errors increases (Muñoz-de-Escalona et al., 2024). With 6-10% of plane crashes being related to ATC errors (Nikšić et al., 2022), avoiding errors in ATC is critical. This can be done by either increasing the amount of ATCOs to keep up with the higher traffic or by enhancing their support systems by machine learning (ML) to take of workload from them (Helmke, Kleinert, Ahrenhold, et al., 2023).

The development of such systems already started in parallel with the development of the first DNNs by using dynamic lattice rescoring to improve ASR for ATC (Shore et al., 2012a). The context for the rescoring comes from an an arrival manager (AMAN), a system which assists ATCOs during arrival. In AcListant, one of the first ATC projects using DNNS, an AMAN is fed with the output of an automatic speech recognition (ASR) system that transcribes spoken ATCO commands to text (Ohneiser, Helmke, et al., 2021). This optimizes the landing sequence provided by the AMAN and takes workload of the ATCO. Since then, there have been many works proposing ASR or NLP solutions for ATC (Chen et al., 2017; Pellegrini et al., 2019; Nigmatulina et al., 2021; Zuluaga-Gomez, S. S. Sarfjoo, et al., 2023; Zuluaga-Gomez, Prasad, Nigmatulina, S. S. Sarfjoo, et al., 2023a). The majority of these works targets however the improvement in given a task in comparison to preceding works. But the challenge of developing support systems for ATC lies however not only in the improvement of their accuracy but especially in the high demand on their reliability. That is why the European Union

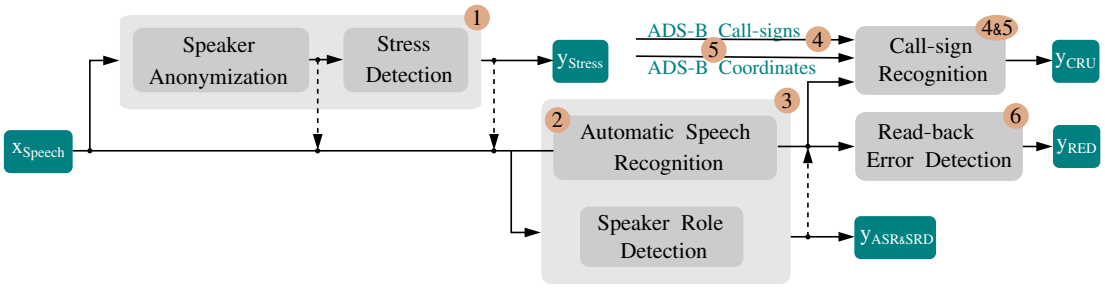


Figure 1.1: Overview of the different topics of this thesis within an ATC speech processing pipeline. Dotted lines represent open research topics.

Aviation Safety Agency (EASA) proposed a guideline to develop trustworthy ML systems for ATC (European Union Aviation Safety Agency, 2021).

The main goal of this thesis to provide analyses and ATC ML solutions, that help to build trustworthy ML systems for ATC. We focus on three mayor key words of the guidelines. The major focus lies on **robustness**. We develop two robust systems for call-sign recognition and understanding (CRU) and a robust method for read-back error detection (RED). Our second focus lies on **explainability**. This is addressed by investigating how lexical and acoustic factors influence ASR for ATC and how the architectural choice of ASR and speaker-role detection (SRD) algorithms influences their performance on ATC dataset. Our third focus lies on **privacy**. An analysis is done on the impact of including privacy measures in ATCO stress detection.

1.2 Contributions

A speech processing pipeline in ATC consists of multiple different modules. For this thesis, we focus on some of the most common tasks, respectively most anticipated tasks in such a pipeline. This maximizes the utility of the results for industrial implementation. Figure 1.1 shows where the contributions of this thesis are located in an ATC speech processing pipeline. In the following, we will elaborate each of the contributions:

1 Stress & Anonymization Stress detection and anonymization are two important topics for ATC. Stress, respectively mental workload is a major factor for ATCO errors as already explained in Section 1.1. In recent years, different methods for predicting stress and workload have been applied, ranging from exterior signals such as traffic density or trajectory uncertainty (Corver et al., 2016) to internal signals like brain activity (Borghini et al., 2020) over pupil size (Muñoz-de-Escalona et al., 2024). There have also been two early works on communication-based workload detection. The first idea is to use communication duration to estimate workload (Uclés et al., 2014). A different approach is taken by Luig et al., who define a roadmap for a speech-based workload monitoring system for ATC, based on different speech features, such as Mel-frequency cepstral coefficients (MFCC) or pitch (Luig et al., 2010). The general problem in ATC is the availability of training data for such a system, since air navigation service providers (ANSPs) cannot share their data, due to privacy concerns. One way to overcome this, is using speaker anonymization, which removes the individual characteristics from the speech. There exist numerous anonymization techniques (Panariello et al., 2024). For the common x-vector based systems exist already multiple evaluations on how to reach the highest privacy (Mohan et al., n.d.).

The focus of Chapter 3, is to evaluate if stress detection in anonymized ATCO speech is possible. Stress detection of ATC speech is a nontrivial task, since ATCOs are trained to always stay calm and keep a similar speech style. But our results show that speech-based stress detection for ATCOs is working. The same holds true for anonymized ATCO speech. In case of a low resource or cross-domain scenario, anonymization leads even to higher detection scores. We suggest that the anonymization acts here as a data augmentation method. When it comes to input features of the tested networks, MFCCs seem to generalize better, while the usage of a log-mel spectrogram leads to better results in the in-domain scenario. Our findings open the gate for further research in ATCO stress recognition, which

could not be conducted before due privacy concerns. The content presented in Chapter 3 is based on:

> Viswanathan, Janaki, **Blatt, Alexander**, Konrad Hagemann, and Dietrich Klakow (Dec. 2022). "Less Stress, More Privacy: Stress Detection on Anonymized Speech of Air Traffic Controllers." In: *Innovation im Fokus* 2, pp. 43–50. URL: https://www.dfs.de/homepage/de/medien/publikationen/internet-fokus2202.pdf?cid=hrf.

As second author, Alexander Blatt has written the paper together with Janaki Viswanathan. Alexander Blatt led the experiment design. All experiments were conducted by Janaki Viswanathan. Konrad Hagemann and Dietrich Klakow provided feedback and advised.

2 **ASR ATC** Pretrained transformer-based ASR models such as Whisper (Radford et al., 2022), wav2vec 2.0 (Baevski et al., 2020) or XLS-R (Babu et al., 2022) show a good performance over a wider range of datasets. This is not the case for ATC datasets. It has been found, that pretraining transformer-based models with unlabeled target domain data can significantly improve their cross-domain performance (Hsu et al., 2021). But even when wav2vec 2.0 or XLS-R are finetuned on multiple ATC datasets, the models reach less than half of their training accuracy when tested on the unseen ATC datasets (Zuluaga-Gomez, Prasad, Nigmatulina, S. Sarfjoo, et al., 2022). This raises the question of the reason for this poor generalization within the ATC domain.

In Chapter 4, we are targeting this question. We investigate how much lexical and acoustic differences between the ATC datasets influence wav2vec 2.0. We do this by evaluating the model on three different datasets with different properties. We find that both, lexical and acoustic differences equally influence the performance of the transformer-based model. We identify that adding Gaussian noise with a specific signal to noise ration (SNR) to clean speech data and performing ASR gives a lower bound for what can be achieved on real-noise data with the same

SNR. We additionally show that adding a target data language model (ML) on top of wav2vec 2.0 does improve the WER, even for noisy data. Analyzes of the wav2vec 2.0 layer outputs show that the feature encoder is agnostic to lexical changes. We identify features in the transformer encoder layers that could indicate a good transferability of a finetuned model. Finally, we provide evidence that the ASR performance on an unseen ATC dataset could be predicted via the ratio between the source-target language model perplexity and the source-target SNR-ratio. This allows to better predict the performance of an ASR model simply by comparing source and target data. The content presented in Chapter 4 is based on:

> **Blatt, Alexander**, Badr M. Abdullah, and Dietrich Klakow (2023). "Ending the Blind Flight: Analyzing the Impact of Acoustic and Lexical Factors on WAV2VEC 2.0 in Air-Traffic Control." In: *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1–8. DOI: 10.1109/ASRU57964.2023.10389646.

As first author, Alexander Blatt designed and conducted all experiments. Badr M. Abdullah helped with the experiment design and the paper writing. Dietrich Klakow provided feedback and advised.

3 **ASR&SRD** In an ATC speech processing pipeline, ASR is one of the most important steps, which allows text-based downstream natural language processing (NLP) tasks such as read-back error detection (RED). Many of those tasks also benefit from or require the knowledge of the speaker role. In RED for example, the read-back of the pilot is compared with the instruction of the ATCO to detect mismatches. Therefore, it is important that the input transcripts of the algorithm are labeled with the speaker roles `PILOT` and `ATCO`. State-of-the-art (SOTA) algorithms for ATC speaker-role detection (SRD) are text-based (Zuluaga-Gomez, S. S. Sarfjoo, et al., 2023). They therefore do not have access to acoustic features. Standard speaker diarization (SD) methods extract however

features from the audio, for example speaker embeddings to differentiate between the different speakers (Park et al., 2022; Dawalatabad et al., 2021). Although SD slightly deviates from SRD, since it aims to label its input with speaker labels, instead of speaker roles, acoustic information is valuable in both cases. Novel SD algorithms take both, lexical and acoustic information as input by simultaneously transcribing and diarizing the audio input (Kanda et al., 2022; Xia et al., 2022), but these architectures are often quite complex.

In Chapter 5 we propose a novel approach for joint speaker-role detection and automatic speech recognition. Our approach builds on (Shafey et al., 2019), but we use simpler transformer-based ASR models instead of transducers. We evaluate our approach against two cascaded approaches, with separated ASR and SRD models. Although the word error rates (WER) of our joint ASR&SRD model are slightly higher than those reached with the best cascaded approach, we show that our joint ASR&SRD model is superior to the cascaded approaches in terms of SRD, while using fewer parameters than the other model. Our model is also consistently better at adding the speaker-role token exactly at the correct position in a sentence. Regarding the cascaded approaches, we can show that acoustic SRD followed by ASR is robust against lexical differences between training and target data, while applying text-based SRD after ASR seems to be the method that gives the lowest WER. This and our joint method can also utilize lexical similarities the best. We can also show that lexical differences between training and target data seem to have a bigger influence on the ASR&SRD models than acoustic differences. Our joint ASR&SRD approach eases the SRD task for ATC since it can be applied to most SOTA ASR architectures and leverage them into an ASR&SRD system. The content presented in Chapter 5 is based on:

Blatt, Alexander, Aravind Krishnan, and Dietrich Klakow (2024). *Joint vs Sequential Speaker-Role Detection and Automatic Speech Recognition for Air-traffic Control*. URL: https://www.isca-archive.org/interspeech_2024/blatt24_interspeech.pdf.

As first author, Alexander Blatt designed and conducted all experiments with the SRD-ASR and Joint architecture. Aravind Krishnan assisted with the experiment design and performed all experiments with the ASR-SRD architecture. Alexander Blatt has written the paper and Aravind Krishnan revised it. Dietrich Klakow provided feedback and advised.

4 **Noise robust call-sign recognition** Since a call-sign is the unique identifier for each flight, it is the most crucial entity in a message from an ATCO to a pilot. Multiple works therefore target call-sign recognition. They can be divided into text-based methods (Pellegrini et al., 2019), and speech-based methods (Nigmatulina et al., 2021). Text-based methods mainly identify the call-sign via named-entity recognition (V. Gupta et al., 2019), thus tagging each word of the call-sign to extract it from the transcript. But there are also other works that use fuzzy string matching between the transcript and the list of call-signs of airplanes in the surrounding area (surveillance call-signs) (Kasttet et al., 2024). Speech-based call-sign recognition can be done by injecting the surveillance call-signs into the language model, and thus improving the call-sign recognition. It should be noted however that a follow-up string matching is needed to really recognize the call-sign. All these methods have the downside that they rely on a good ASR prediction, otherwise the matching does not work, respectively just parts of the call-sign are extracted via tagging. Furthermore, half of the methods heavily depend on the surveillance data and only work if the call-sign is present in the surveillance data.

In Chapter 6 we propose a new call-sign recognition and understanding system (CRU) and data augmentation method. Our data augmentation method allows us to produce training data from unseen airports just by using the surveillance call-signs from the new airport and we can additionally simulate low SNR conditions. The results show that this makes our system more robust against ASR errors. Our model is able to extract a call-sign from a transcript and convert it in a standardized format in one step. In contrast to previous architectures, our proposed model works with, but importantly also without surveillance call-signs.

We investigate our system in different noise conditions and also variate the quality of the surveillance call-signs and can show that system is robust over a wide operation range. This closes the gap further between research and an operational speech-processing pipeline for ATC. The content presented in Chapter 6 is based on:

> **Blatt, Alexander**, Martin Kocour, Karel Veselý, Igor Szöke, and Dietrich Klakow (2022). "Call-Sign Recognition and Understanding for Noisy Air-Traffic Transcripts Using Surveillance Information." In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8357–8361. DOI: 10.1109/ICASSP43922.2022.9746301.

As first author, Alexander Blatt designed and conducted all experiments. Martin Kocour, Karel Vesely and Igor Szöke provided the audio transcripts for the experiments that were labeled with call-signs by the first author. Dietrich Klakow provided feedback and advised.

5 **Call-sign recognition for edge cases** Although we mentioned above that there exist several methods for call-sign recognition. None of them is robustified for all the different edge cases that happen in ATC. This is a problem, since the European Union Aviation Safety Agency specifically states in its guides that systems that are build for operation must be tested and also be robust at edge cases (European Union Aviation Safety Agency, 2021).

In Chapter 7, we identify three main edge cases that influence text-based call-sign recognition: Firstly, high WER transcripts that are due to noisy speech. Secondly, clipped transcripts that are due to a wrong usage of the push-to-talk button by an ATCO and thirdly completely missing transcripts due to low SNR values. We optimize or SOTA CRU system from above for these edge-cases and show that this improves its performance even further over a broad operational range. We additionally introduce two new architectures. One architecture, called CallSBERT, which is significantly smaller and faster than SOTA CRU models,

while only lacking a fraction of their performance. The other architecture is called call-sign-command recovery model (CCR) and can be combined with different CRU architectures and enables them to also utilize plane coordinates for CRU. This shift to a multimodal input allows to even recognize call-signs, when an ASR system does not produce a transcript. This further robustifies CRU and brings it a step closer to be used in operation. The content presented in Chapter 7 is based on:

> **Blatt, Alexander** and Dietrich Klakow (2024). *Utilizing Multimodal Data for Edge Case Robust Call-sign Recognition and Understanding*. arXiv: 2412.20467 [cs.CL]. URL: https://arxiv.org/abs/2412.20467.

As first author, Alexander Blatt designed and conducted all experiments. Dietrich Klakow provided feedback and advised.

6 Read-back error detection Air-traffic controllers issue up to one command every two minutes in addition to their monitoring work (Lehouillier et al., 2014). This accumulates in an eight hour shift to 240 commands. These commands have to be read back by the pilot, to ensure the correct understanding of the command. A not correctly read back command is a read-back error. If not detected by the ATCO (hear-back error), the miscommunication can lead to incidents (Yang et al., 2023). At the example of clearance commands, it has been shown, that this miscommunication only happens in 0.1% of the commands uttered. (Morrow et al., 1993). Projected to the 240 commands, this still leads to a wrong pilot behaviour every four days per controller, which is not negligible. To support the ATCO in detecting all read-back errors, read-back error detection (RED) systems have been in the focus of research in the last decade.

In recent years, different architectures of RED methods have been proposed. One idea is to extract a semantic vector representation from the ATCO and the pilot transcript and feed them as input into a classifier network. Long short-term memory networks (LSTMs) (JIA et al., 2018a) and convolutional neural networks

(Cheng et al., 2018) have both been shown to be able to extract meaningful representations. Another ML-based approach utilizes BERT (Devlin, M.-W. Chang, et al., 2018) to differentiate between `read-back` or `no read-back` (Helmke, Ondřej, et al., 2022).

In Chapter 8, we suggest the first purely machine-based algorithm that is able to detect multiple read-back error classes. We identify a two-stage training, that first fine-tunes the RED classifier on noisy data and then on clean+augmented data as the superior method for handling the high class imbalance in RED. We can show that our method is also robust when confronted with unseen airspaces, which opens the door for RED support systems for every airport, even without available training data. The content presented in Chapter 8 is based on:

> Bashyam, Lakshmi Rajendram, **Blatt, Alexander**, and Dietrich Klakow (2023). "Enabling Noisy Label Usage for Out-of-Airspace Data in Read-Back Error Detection." In: *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1–8. DOI: 10.1109/ASRU57964.2023.10389759.

As second author, Alexander Blatt led the paper writing and experiment design. All experiments were conducted by Lakshmi Rajendram Bashyam who helped writing the paper and participated in the experiment design. Dietrich Klakow provided feedback and advised.

1.3 Additional publications

In addition to the contributions outlined above, during my PhD, I have contributed as a coauthor to the following publications and preprints which are beyond the scope of this thesis:

Kocour, Martin, 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 (Dec. 2022). "Automatic Processing Pipeline for Collecting and Annotating Air-Traffic Voice Communication Data." In: *Engineering Proceedings* 2.1, p. 8. DOI: 10.3390/engproc2021013008.

Rigault, Mickaël, Claudia Cevenini, Khalid Choukri, Martin Kocour, Karel Veselý, Igor Szoke, Petr Motlicek, Juan Pablo Zuluaga-Gomez, **Alexander Blatt**, Dietrich Klakow, Allan Tart, Pavel Kolčárek, and Jan Černocký (2022). "Legal and Ethical Challenges in Recording Air Traffic Control Speech." In: *Proceedings of the Workshop on Ethical and Legal Issues in Human Language Technologies and Multilingual De-Identification of Sensitive Data In Language Resources within the 13th Language Resources and Evaluation Conference* June, pp. 79–83. URL: https://aclanthology.org/2022.legal-1.14.

Zuluaga-Gomez, Juan, Karel Veselý, Igor Szöke, **Alexander Blatt**, Petr Motlicek, Martin Kocour, Mickael Rigault, Khalid Choukri, Amrutha Prasad, Seyyed Saeed Sarfjoo, Iuliia Nigmatulina, Claudia Cevenini, Pavel Kolčárek, Allan Tart, Jan Černocký, and Dietrich Klakow (2022). "ATCO2 corpus: A Large-Scale Dataset for Research on Automatic Speech Recognition and Natural Language Understanding of Air Traffic Control Communications." In: pp. 1–29. arXiv: 2211.04054. URL: http://arxiv.org/abs/2211.04054.

Kocour, Martin, Karel Veselý, **Alexander Blatt**, Juan Zuluaga Gomez, Igor Szöke, and Jan Černocký (2021). "Boosting of contextual information in ASR for air-traffic call-sign recognition." In: pp. 2993–2997. DOI: 10.21437/Interspeech.2021-1619.

2

Background

Contents

This chapter contains the necessary background information for the following chapters. Despite the purpose of self-containment, we also give here additional information that goes beyond the scope of the main chapters. Firstly, we introduce

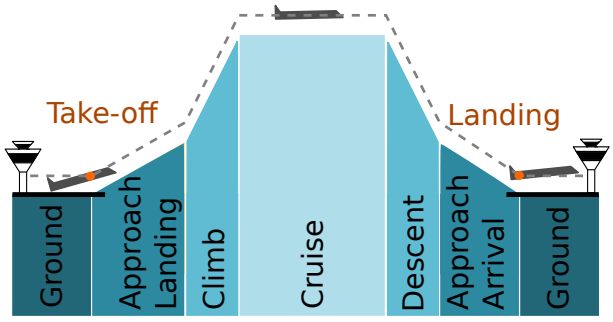


Figure 2.1: Different flight phases of a plane.

the basics of air traffic control in Section 2.1 to improve the understanding of the target domain of this thesis. Next, we shift to the ML-based background, starting with the ML fundamentals in Section 2.2. This is followed by Section 2.3, where we discuss the different ML architectures used in this thesis. Lastly, in Section 2.4, we introduce the ML tasks that are targeted within the following chapters.

2.1 Air-traffic control

Air-traffic control (ATC) ensures safe flight operation in all flight phases shown in Figure 2.1. The most visible ATC structure at each airport is the control tower. The **air-traffic controllers** (ATCOs) in the control tower, the **tower controllers**, are mainly responsible ground control and take-off. This involves e.g. giving instructions for taxiing to the runway or giving take-off clearances. Similarly, during landing, a landing clearance and taxi commands are given. In the approach, climb and descend zone, an **approach controller** handles an airplane. This includes giving specific heading commands to an airplane or assigning flight levels. In contrast to the tower controllers, the approach controllers do not need to have visual contact with the planes, therefore they can be located outside of the airport. The same holds true for area or **en-route controllers** which handle the cruise phase of a plane. Since there occur not many direction changes during the cruise phase, the airspaces controlled by an en-route controller are much larger than the airspaces controlled by approach and tower controllers. The Maastricht

Upper Area Control Centre (MUAC)[1] controls for example Belgium, Luxembourg, the Netherlands and the northwest of Germany. In this work, we focus mainly on approach and tower communications. An exhaustive description of the entire ATC and the air-traffic management (ATM) structure can be found at SKYbrary[2].

2.1.1 Communication modalities

Depending on the flight phase, different types of communication are used. During the en-rout phase, the commands issued by the controller are not as time-critical as the commands during landing. Therefore, many European en-route control centers are using the **controller–pilot data link communications** (CPDLC) system. This text-based system solves two problems of the traditional speech-based communication. During the normal **very high frequency** (VHF) band communication, an ATCO controls several pilots via a 50 KHz band between 108 and 137 MHz (Raab et al., 2002). Since all pilots controlled by one ATCO are tuned in on the same frequency, they all hear the ATCO issuing commands, despite the command being just addressed to one pilot. This not only affects the attention of the pilots, but can also lead to misunderstandings. Additionally one pilot could accidentally override another pilot speaking, if they try to communicate at the same time. Text-based communication avoids these problems (Ďurčo et al., 2017).

During, landing, take-off and approach, fast reaction times are needed, voice communication is here still superior to text-based communication. However, voice-based communication is prone to errors due to miscommunications, noise, and different accents (Tiewtrakul et al., 2010). To avoid errors, ATC communication is based on a strict phraseology.

1 https://www.eurocontrol.int/muac
2 https://skybrary.aero/

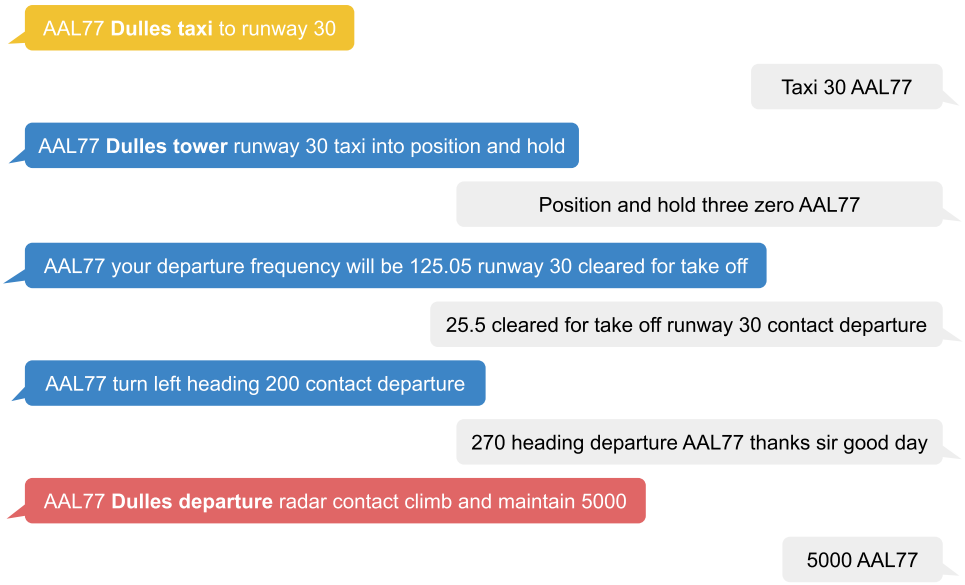


Figure 2.2: ATC conversation during the take-off of American seven seven (AAL77) and Dulles **taxi** (yellow), **tower** (blue) and **departure** (red) (Gregor, 2001).

2.1.2 Phraseology and important entities

The goal of the **ATC phraseology** is to standardize the communication and avoid misunderstandings. An example of the applied phraseology in an ATC communication is shown in Figure 2.2. The messages of the different ATCOs are shown on the left, and the answers of the pilot of flight `AMERICAN SEVEN SEVEN` (AAL77) are shown on the right. Each ATCO message starts with the **call-sign** AAL77. This is a unique identifier for the flight. Since several pilots can be tuned into the same frequency, as described in the previous section, it is crucial to make clear which pilot is addressed by the ATCO. The short form AAL77, standardized by the International Civil Aviation Organization (ICAO)[3] consists of the three letter long airline identifier AAL, which is the short form of `AMERICAN` and stands for American Airline. The airline identifier is followed by the call-sign number, an alpha-numerical code. For vocalization of the call-sign number, the ICAO phonetic

3 https://www.icao.int

alphabet[4] is used. The call-sign DLH1LT is therefore vocalized as `LUFTHANSA ONE LIMA TANGO`.

The call-sign is followed by the **command**. Command examples from Figure 2.2 are `CLIMB`, `TAXI TO`, `TURN LEFT` and `CLEARED FOR TAKE OFF`. The command specifies what the pilot should do next and is typically followed by a **value**. The value further refines the command. In `TAXI TO RUNWAY 30`, the value `RUNWAY 30` specifies to which runway the plane must taxi. The same holds true for `TURN LEFT HEADING 200`, where the `HEADING 200` makes clear by how far the plane should turn to the left. In the ATCO2 project [5], we have labeled hundreds of ATC communication transcripts with the main ATC entities call-sign, command and values, resulting in the ATCO2 corpus (Zuluaga-Gomez, Veselý, Szöke, et al., 2022). This corpus can for example be used to train a neural networks for ATC named-entity recognition (NER).

The sentence structure of the pilot utterances differs from the ATCO utterances. For safety reasons, a pilot is obligated to read back the command uttered by the ATCO. The read-back usually starts with the command and the value and is terminated by the call-sign. If the read-back is incorrect, like the heading command in Figure 2.2, the ATCO can identify the misunderstanding and repeat the instruction until the pilot gives the correct read-back. This measure, if conducted correctly, prevents accidents and incidents (Alharasees et al., 2023).

2.1.3 Surveillance Technologies

The term **surveillance information** in ATC is often still associated with **radar** information, but the usage of radar to determine the position of airplanes has disadvantages. In high traffic airspaces, for example Frankfurt Airport, the radar display can get crowded. Furthermore, the radar system only provides positional information. Since radar is also highly affected by obstacles between the radar

4 https://skybrary.aero/articles/icao-phonetic-alphabet
5 https://www.atco2.org/

antenna and the plane, which unfortunately also includes clouds, the system is not very reliable. Therefore, Automatic Dependent Surveillance–Broadcast or in short **ADS-B** has been introduced and is already mandatory for many aircrafts in Europe and America (Rekkas, 2014). There are two components of an ADS-B system: ADS-B IN (receiving) and ADS-B OUT (transmission). Aircrafts are broadcasting their position, velocity, status information and an their aircraft identifier (call-sign) every second via ADS-B OUT. Other aircrafts, ATC providers, respectively everyone with an ADS-B IN system is capable of receiving that information. Aircraft-to-aircraft information transmission enables planes to detect other nearby planes, increasing situational awareness, which prevents incidents (Kožović et al., 2023). The quality of the information is also greatly enhanced, in comparison to radar, by the additional velocity information. The call-sign in the ADS-B data also allows us to link the positional information to ATC speech, respectively their transcripts. This leverages the ADS-B data to an important surveillance information for ATC speech related tasks as shown in Chapter 6 and Chapter 7. This is facilitated by the OpenSky Network[6] (OSN) database (Schäfer et al., 2014). This database stores ADS-B information from the whole world and allows to extract the ADS-B data from the target airspace via coordinates and time stamps.

2.2 Machine learning fundamentals

Finding a mathematical function f that generates the outputs y_n given the inputs $\boldsymbol{x}_n$ for $\forall n = 0, 1, 2, ..N$ in the way that

$$y_n = f(\boldsymbol{x}_n) \tag{2.1}$$

6 OSN Homepage: https://opensky-network.org/

can be a non-trivial task. Training **neural networks (NN)** allows to iteratively find or closely approximate such a function. A NN can be described as a function f_θ, with θ as trainable network parameters. The network maps x_n to y_n^p:

$$y_n^p = f_\theta(\boldsymbol{x}_n) \ . \tag{2.2}$$

The goal of training or fine-tuning a neural network is to find the optimal network parameters $\boldsymbol{\theta}_{opt}$ so that $y_n = y_n^p$, which is equivalent to $f_{\boldsymbol{\theta}}(\boldsymbol{x}_n) = f(\boldsymbol{x}_n)$. This goal can be achieved by minimizing a loss function $\mathcal{L}(\theta)$ that measures the difference between the target and predicted outputs:

$$\theta_{opt} = \arg\min_\theta \ \mathcal{L}(y_n - y_n^p) \tag{2.3}$$

$$= \arg\min_\theta \ \mathcal{L}(y_n - f_{\boldsymbol{\theta}}(\boldsymbol{x}_n)) \ . \tag{2.4}$$

A common loss function is e.g. the **mean squared error (MSE)**,

$$MSE = \frac{1}{n}\sum_{n=1}^{N}(y_n - y_n^p)^2 \ , \tag{2.5}$$

which is used in regression problems. Choosing the right loss function for each task is crucial for the model performance (Q. Wang et al., 2022). Instead of finding a closed form solution for Equation 2.4, the optimal parameters can be found iteratively via training the network with **gradient descent**. The training process can be described as follows:

1. Initializing of the network with (random) parameters $\boldsymbol{\theta}_0$
2. Calculating the gradient of the loss function $\nabla_{\boldsymbol{\theta}_t}\mathcal{L}$ at step t
3. Updating of the network parameters in negative gradient direction weighted with the learning rate η via:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\nabla_{\boldsymbol{\theta}_t}\mathcal{L} \ . \tag{2.6}$$

4. Starting again at step 2

The goal of gradient descent is to update the parameters so that the loss gets smaller, therefore the update is done in negative gradient direction with respect to the parameters. For the standard gradient descent, the gradient is calculated across the entire training dataset at each step. To reduce the computational burden, the **stochastic gradient descent (SGD)** (Amari, 1993) can be used, which calculates the gradient over a subset of the data, named **batches**. SGD variants that improve the convergence behaviour are for example RMSProp or **ADAM** (Kingma et al., 2015). The training process can be terminated if the network results do not improve anymore. This can be either measured via loss decrease, or with another target metric, for example the **word error rate (WER)** (Klakow et al., 2002). The termination can be implemented with **early stopping** (Caruana et al., 2001), which terminates the training if the model results do not improve for a certain number of full train dataset iterations **epochs**.

2.3 Machine learning architectures

In this chapter, we introduce the basics of the architectures used in this work. For the sake of understanding, we focus on the most important architectures and architecture layers.

2.3.1 Feedforward neural networks

Feedforward neural network (FNN), shown in Figure 2.3, are one of the first neural networks used and rely on a simple architecture without loops. The parameters for the layer i are:

$$\boldsymbol{a}_i = \boldsymbol{W}_i \boldsymbol{x} + \boldsymbol{b}_i \tag{2.7}$$

$$\boldsymbol{h}_i = \phi(\boldsymbol{a}_i) \tag{2.8}$$

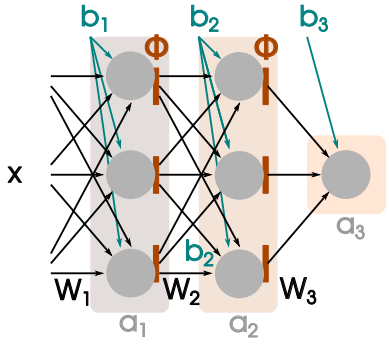


Figure 2.3: Feedforward neural network (FFN) with two three neuron layers and one final output layer with one neuron.

with $\boldsymbol{x} \in \mathbb{R}^{l \times 1}$ either being the network input or the output of the previous layer h_{i-1}. The network parameters consists of the layer weight matrix $\boldsymbol{W}_i \in \mathbb{R}^{d \times l}$ and the layer bias vector $\boldsymbol{b}_i \in \mathbb{R}^{d \times 1}$. The **activation function** ϕ in an FNN introduces non-linearity and therefore allows to find more complex mappings between input and output. Common activation functions are **relu**, sigmoid or tanh (Dubey et al., 2022; Apicella et al., 2021).

The connectivity between subsequent layers is so high that each **neuron** gets its input from all neurons of the previous layer as shown in Figure 2.3. This allows the network to explore various combinations to find the optimal mapping between input and output. Therefore, FNNS can be found in numerous complex architectures in the form of **fully connected layers** (Devlin, M.-W. Chang, et al., 2018; Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Aidan N. Gomez, et al., 2017b), and they are generally used as the final classification layer for a network (Basha et al., 2020). The drawback of the high connectivity of FNNs is the amount of network parameters that needs to be stored. This results in a high memory demand and long training times.

2.3.2 Convolutional neural networks

While an FNN contains many parameters because of its connectivity, a **convolutional neural network** (CNN) is based on a different approach. The trainable

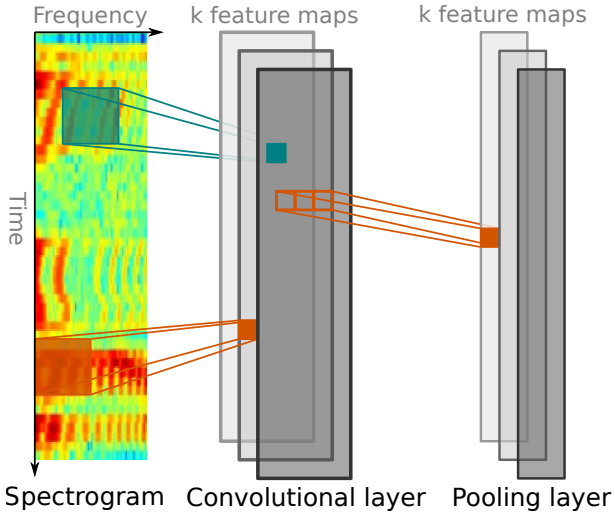


Figure 2.4: Convolutional neural network (CNN) architecture consisting of a convolutional and a pooling layer.

parameters of a **convolutional layers** are convolutional **kernels**, which are matrices, with the dimension $\boldsymbol{M} \in \mathbb{R}^{m \times n \times k}$ for 2D and 1D convolutions. n and m define the size of the kernel and k equals the number of different kernels learned. The convolutional layer in Figure 2.4 shows for example quadratic 2D kernels of the size $n = m$. For 1D convolutions, either n or m equals 1. 1D convolutions can only detect one-dimensional features, therefore they are for example used for 1D inputs like single sensor outputs (Kiranyaz et al., 2021). 2D convolutions are on the other hand used for 2D inputs like pictures or spectrograms (Z. Li et al., 2022). Depending on the features learned, a kernel can be used to detect for example high frequency features, like sharp edges or other features. In a CNN, the kernel is shifted over the image and the dot product between the kernel and the input is calculated at each position to identify the kernel-specific features all over the input. This makes this network architecture very efficient because even for a large input, only a significantly smaller kernel function needs to be learned. Stacking multiple convolutional layers increases the field of view of the upper kernels with respect to the input. This allows the extraction of bigger, respectively high level features. To detect different features, k different kernels are learned instead of

a single one. Convolving each of these kernels with the input results in the k **feature maps** shown in Figure 2.4.

The second most important layer in a CNN after the convolutional layer is the **pooling layer**. The pooling layer decreases the dimension of the feature maps by pooling features in a defined window of the previous layer. Figure 2.4 shows for example a 1×3 pooling kernel. Different pooling methods, like mean, max or min pooling, give different weights to the original features during pooling (Zafar et al., 2022). Max pooling extracts for example the dominant feature of the pooling window. If a CNN is used as a standalone classifier, the final layers are mostly fully connected layers (Basha et al., 2020). In modern speech recognition architectures, CNNs are mostly used as basal feature extraction layers (Baevski et al., 2020).

2.3.3 Transformers

The introduction of **transformers** by (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Aidan N Gomez, et al., 2017a) has led to a paradigm shift in multiple language, speech and vision tasks. The original transformer architecture is shown in Figure 2.5 and consists of an encoder and a decoder block. In contrast to recurrent networks like long short-term memory (LSTM) networks (Greff et al., 2017), the input sequence is not fed in sequentially, from start to end, put in parallel. This allows a much faster processing. The sequential order of the input sequence is preserved via the addition of positional encoding to the input embedding. This modified input is fed into the transformer **encoder** block, specifically the first **multi-head attention** layer. Each head of the multi-head attention layer learns to ”give attention” to specific positions of the input (more details about the attention mechanism can be read-up in (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Aidan N Gomez, et al., 2017a) since attention is not in the focus of this work). In a text-based input, one head could for example learn to focus on the beginning of the sentence, to find the subject of the sentence. The multi-head

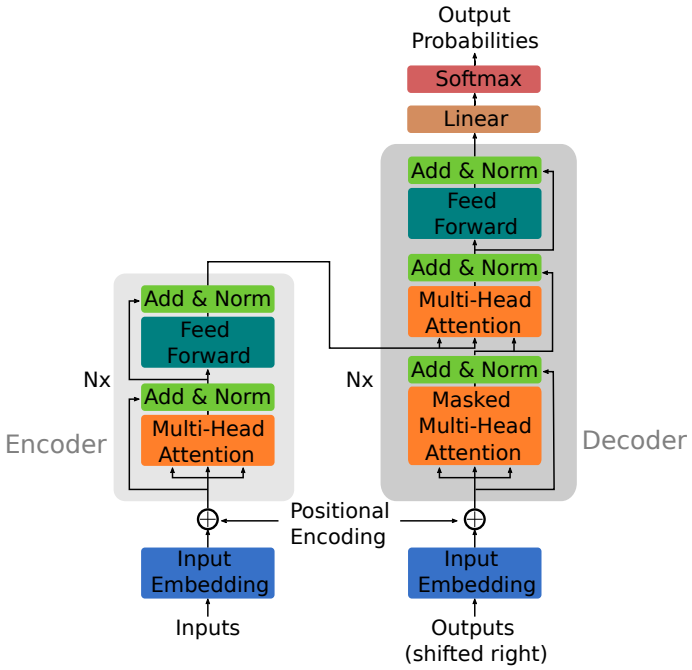


Figure 2.5: Transformer architecture consisting of an encoder and a decoder block as introduced in (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Aidan N Gomez, et al., 2017a).

attention layer is followed by a residual connection that allows the input to bypass the multi-head attention layer and a layer normalization step that stabilizes the training process (L. Huang et al., 2023). This is followed by a position-wise 3 layer FNN as described in Section 2.3.1 and again an "Add&Norm" layer. This structure is repeated N times as shown in Figure 2.5.

The **decoder** has a similar structure, but it also takes in the output of the encoder as input. The masked multi-head attention allows the decoder only to attend to positions that are "left" of the the current position it should predict. The decoder therefore predicts the next token based on the encoder output and the previously predicted output. This makes the transformer decoder an **autoregressive** model, while the transformer encoder is a **bidirectional** model. Bidirectional architectures based on the transformer encoder are **BERT**, ALBERT or wav2vec 2.0 and autoregressive models are GPT1-4, LLaMa or Bart (Kalyan et al., 2021). While the bidirectional models are mainly used for natural language

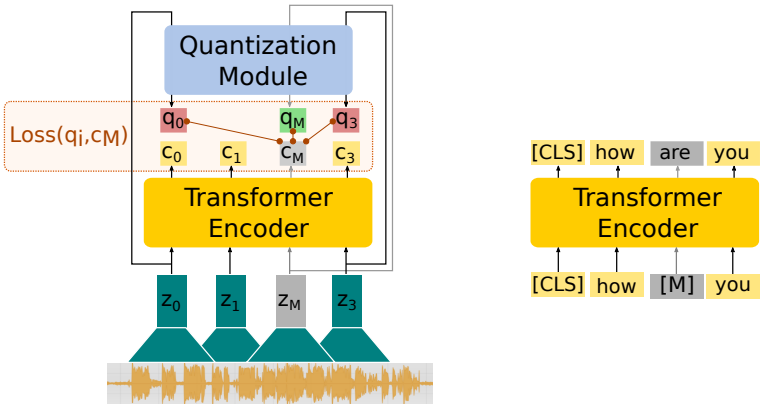


Figure 2.6: The transformer encoder-based architectures wav2vec 2.0 (left) and BERT (right) while pretraining via masking.

processing (NLP) tasks, autoregressive models are used for sequence-to-sequence tasks like translation or question answering.

In Chapter 6 and Chapter 7, we use the full transformer architecture for call-sign recognition. In Chapter 7, we use additionally BERT, shown in Figure 2.6 (right) as bidirectional model for call-sign recognition. BERT is also used in Chapter 8 for read-back error detection. **BERT** or bidirectional encoder representations from transformers is a pretrained transformer encoder-based architecture. BERT is **pretrained** on unlabeled data via **masked language modeling** (MLM) by masking 15% of the input tokens (words) and predicting them as output. This pretraining step is the key of the success of transformer architectures, since it does not require labeled data and can therefore be performed on large databases like Wikipedia or web Common Crawl data (Kalyan et al., 2021). Pretraining allows the model to adapt to a certain language or domain, like for example the air-traffic control domain. **Finetuning** BERT on the final task, e.g. call-sign recognition, can then be done on a smaller labeled corpus, by adding a classification head, based on a FNN on top of the encoder.

Another encoder based architecture is **wav2vec 2.0** (Baevski et al., 2020), shown in Figure 2.6 (left). Wav2vec 2.0 is used in Chapter 4 for ASR and in Chapter 5 for speaker role detection. During fine-tuning, wav2vec 2.0 also consists of an encoder-FNN stack, similar to BERT. But since wav2vec 2.0 gets audio as

input, it possesses a CNN based feature encoder that converts the continuous speech signal into a discrete signal. This makes the pretraining step, shown in Figure 2.6 (left) more complicated. The CNN produces feature vectors $\boldsymbol{Z}_0, \boldsymbol{Z}_1, \boldsymbol{Z}_2..\boldsymbol{Z}_T$ for every 20 ms of speech. These feature vectors are fed into the transformer encoder and the quantization module. The transformer encoder produces the projected context vectors $\boldsymbol{c}_0, \boldsymbol{c}_1, \boldsymbol{c}_2..\boldsymbol{c}_T$ out of the input. The quantization module converts the feature vectors to quantization vectors $\boldsymbol{q}_0, \boldsymbol{q}_1, \boldsymbol{q}_2..\boldsymbol{q}_T$. These quantization vectors are sampled from a finite codebook, which allows to produce a finite vocabulary, similarly to a language processing task (A detailed description of the quantization module, which is beyond the scope of this chapter, can be found in (Baevski et al., 2020)). During pretraining, a feature vector $\boldsymbol{Z}_M$ is masked and the transformer encoder is trained by contrastive loss to distinguish between the correct quantization vector ($\boldsymbol{q}_M$ in Figure 2.6) and the distractors ($\boldsymbol{q}_0$ and $\boldsymbol{q}_3$ in Figure 2.6). During the contrastive loss calculation, the cosine similarity between the quantization vectors and the projected context vector at the masked position ($\boldsymbol{c}_M$) is calculated with the goal to reduce the distance between $\boldsymbol{c}_M$ and $\boldsymbol{q}_M$ and increase the distance between $\boldsymbol{c}_M$ and $\boldsymbol{q}_0, \boldsymbol{q}_3$. Roughly 50% of the feature vectors are replaced with the masked feature vector $\boldsymbol{Z}_M$ and for each masked position 100 distractors are sampled over the input sequence positions during pretraining.

The quantization module is dropped before finetuning as already explained above. For ASR, a language modeling head on top of the transformer encoder is used for connectionist temporal classification (CTC) (Graves, Fernández, et al., 2006a). **CTC** is an algorithm that allows to train a network without knowing the alignment between an input sequence $x_1, x_2, ..., x_n$ and an output sequence $y_1, y_2, ..., y_n$. Since the speech rate varies between speakers and even for one speaker, overcoming the alignment problem is crucial. Figure 2.7 shows how CTC overcomes this problem by first continuously predicting a letter for a fixed timeframe. In the next step, repeated characters, which allow to accommodate to different speech rates, get merged. In the last step, the blank token ϵ, which allows silence but also

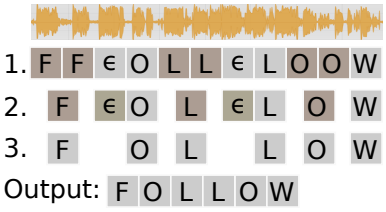


Figure 2.7: CTC algorithms steps, After the initial sequence prediction (1), repeated characters are merged, followed by the removal of the blank token.

double consonants, is removed. An in-depth view of the CTC is given in (Graves, Fernández, et al., 2006a).

2.3.4 Multimodal networks

All models introduced in the previous sections are based on one input. In reality, models often rely on multiple input modalities (Sleeman et al., 2021). These **multimodal networks** can utilize speech, text, radar, sensor data or other modes to either enhance prediction accuracy or increase robustness, if one of the input modalities is not available, or the measurements are noise (Ngiam et al., 2011). Improving ASR by including visual features is an example of this. In Chapter 7, we use ADS-B information to improve the robustness and accuracy of call-sign recognition and understanding (CRU) in edge cases. Depending on when the information of the different modes is merged, one can differ between the two main scenarios shown in Figure 2.8 (Sleeman et al., 2021). In the **late fusion** approach, a model is trained for each modality, for example a separate speech and text classifier. This gives the benefit that already finetuned single-modality models can be utilized. The **early fusion** approach relies on training a model based on both modalities. Just a feature extractor is used to preprocess each modality, this could be for example a pretrained wav2vec 2.0 model (speech) or a pretrained BERT model (text). This approach allows the model to be optimally adapted to both modalities.

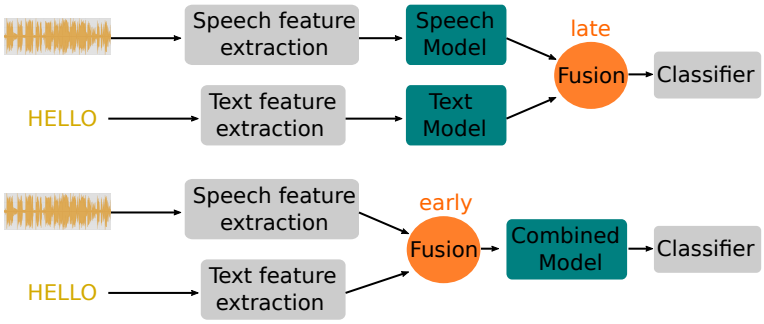


Figure 2.8: Different fusion approaches based on the point of fusion. In late fusion (top), individual models are trained for each modality and in early fusion (bottom) a single model is trained for both modalities.

Our CRU architecture in Chapter 7 is a combination of both approaches. The joining of the features themselves, can be realized via a simple concatenation, or a DNN based feature merging.

2.4 Machine learning tasks

This section introduces speech and language processing related machine learning tasks, that are relevant for the following chapters.

2.4.1 Named-entity recognition

named-entity recognition (**NER**) is a common **natural language processing** (NLP) task. The goal is to identify named entities in a text document. Common named entities are for example LOCATION, PERSON or ORGANIZATION (Nadeau et al., 2007). These entities vary however depending on the domain. Important named entities for ATC are CALL-SIGN, COMMAND and VALUE as described in Section 2.1.2. A good overview of NER algorithms and datasets is given in (J. Li et al., 2022). There exist also multiple tagging schemes for named entities. A widely used scheme is the **IOB scheme** introduced by Ramshaw et al.

Figure 2.9: IOB tagged sentence with the named entities call-sign (Cal), command (Com) and value (Val).

(Ramshaw et al., 1999). In this scheme, a token, respectively word is marked to be either at the beginning (B), inside (I) or outside (O) of a named entity. An IOB labeled ATCO instruction is shown in Figure 2.9. Despite the IOB scheme, there exist also other schemes, for example the IOBES scheme which also explicitly tags the end of a named entity. This additional information can lead to a higher accuracy, respectively F1 score (Alshammari et al., 2021).

The **F1 score** is a measure, used in classification, that combines **precision**, defined as

$$precision = \frac{tp}{tp + fp} \tag{2.9}$$

and **recall**, defined as

$$recall = \frac{tp}{tp + fn} \tag{2.10}$$

in one metric via:

$$F_1 = 2\frac{precision \cdot recall}{precision + recall}. \tag{2.11}$$

With tp beeing the true positive, fp beeing the false positive and fn beeing the false negative classified instances. All three measures, F1, recall and precision range from 0 (low) to 1 (high). Assuming a binary NER task, where a token either belongs to a named entity or not, precision measures how many of the tokens that are labeled as part of a named entity are correct. Recall on the other hand measures how many of all tokens that are part of a named entity are detected by the classifier. In a scenario with a highly unbalanced class distribution, the F1 score is preferable over the accuracy, which overfits the majority class

In this work, we focus on detecting the call-sign as named entity in ATCO, respectively pilot transcripts. We go however a step further and also automatically convert the detected call-sign to the standard ICAO format as shown in Chapter 6. We therefore call this task **call-sign recognition and understanding** (**CRU**).

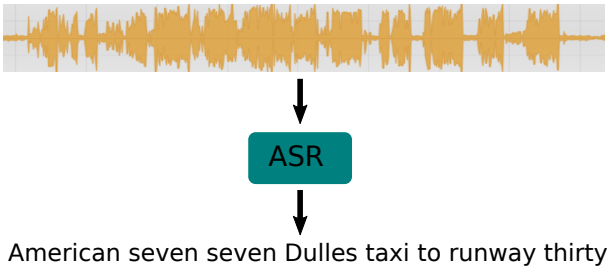


Figure 2.10: ASR converts an audio recording to text.

2.4.2 Automatic speech recognition

Automatic speech recognition (**ASR**) or speech-to-text (STT) systems transcribe an audio recording to text as shown in Figure 2.10. An introductory overview of ASR models and techniques can be found in (Malik et al., 2021). Traditional ASR models are based on a cascaded structure. An acoustic model, that maps the audio input to phonemes, is followed by a language model, which maps the phonemes to words (Derouault et al., 1986). The important information of the audio input lies within the frequency range of the human voice which ranges roughly from 50-20000 Hz (Monson et al., 2014). Within this range, humans are able to differentiate pitch differences better for lower than for higher frequencies, which is the basis of the **mel scale** (Vergin et al., 1999):

$$mel = 2595 \log_{10}(1 + \frac{f}{700}). \tag{2.12}$$

This scale is embedded in the calculation of the **mel-frequency cepstral coefficients** (**MFCCs**) (Zheng et al., 2001). The mel scale is used in the MFCC calculation to generate a filterbank, which contains vectors, that are evenly spaced in the mel, but not in the frequency scale. These vectors are used to downsample the frequency range of a spectrum. After calculating the logarithm of the resulting filterbank coefficients, they are uncorrelated via discrete cosine transform (DCT) to get the MFCCs. The MFCCs contain therefore the information of a spectrogram in a condensed form. Since the whole frequency range is condensed in a small amount of coefficients, this number needs to be sufficiently high to preserve

the relevant frequency information. Despite the dependence on the language, 25 coefficients are a good general choice (Hasan et al., 2021).

In traditional ASR systems, MFCCs are the input of the **acoustic model**. The acoustic model maps the MFCCs to phonemes, more precisely to a probability distribution over phonemes, respectively characters for chunks (windows) of the input data. The **language model**, which is trained on text data, scores the output of the acoustic model based on the statistics learned during training. State-of-the-art transformer-based **large language models** (**LLMs**) like GPT-4 are trained on a hundreds of GB up to TB of text data, reaching human-level performance in several tasks (OpenAI et al., 2023).

Modern ASR systems rely on a ML-based end-to-end architectures, first introduced in 2014 (Graves and Jaitly, 2014). The model input can consists of raw audio data, as it is the case for wav2vec 2.0 (Baevski et al., 2020), introduced in Section 2.3.3. But one of the most successful end-to-end models, whisper, still relies on the mel scale by using the log-mel spectrogram as input (Radford et al., 2022). The breakthrough of these models lies within the capability to be pretrained on a massive amount of unlabeled speech data. End-to-end ASR systems are in most cases trained via CTC loss, which is further explained in Section 2.3.3.

To evaluate the performance of an ASR method, the **word error rate** (**WER**) is used. The WER measures the distance between the reference transcript and the ASR transcription by adding up the changes between the reference and the prediction:

$$WER = \frac{I + D + S}{N}. \tag{2.13}$$

The sum of the inserted I, deleted D and substituted S words in the prediction is divided by the sum of all words in the reference N.

The ATC domain poses a challenge for ASR models due to noisy recordings, accented and multilingual speech and domain specific vocabulary. We further investigate this in Chapter 4 and Chapter 5.

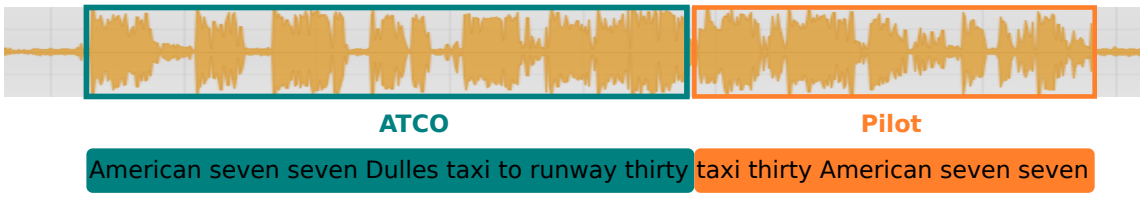


Figure 2.11: Labeling of a utterance and its transcript with speaker role labels.

2.4.3 Speaker Role Detection

The goal of **speaker role detection** (**SRD**), also known as speaker role recognition, is to detect the speaker role belonging to a word sequence, respectively a portion of an audio recording. Figure 2.11 shows this at the example of the speaker roles `ATCO` and `Pilot`. SRD differs from diarization, which can be best explained by looking at the tagging. In diarization, there consists a speaker-tag for each speaker, in SRD, the speaker role tag is not bound to a single speaker. This makes SRD for example interesting for dialog-based tasks that are based on specific speaker roles like: `doctor - patient` (Flemotomos et al., 2020), `interviewer - guest` (Bellagha et al., 2020) or, as mentioned above, `ATCO - Pilot` (Zuluaga-Gomez, S. S. Sarfjoo, et al., 2023). In case of SRD tagged speech, speaker role specific ASR algorithms can be applied, which leads to optimal adaptation to the role specific vocabulary. Similarly, for SRD tagged text data, speaker role specific NLP technologies, like NER can be applied. Since diarization and SRD are closely related, algorithms and NNs for diarization can often also be used for SRD. Therefore, these two diarization reviews give also a good overview of SRD architectures: (Serafini et al., 2023; Park et al., 2022).

We investigate SRD in the ATC domain, with the speaker roles `ATCO` and `Pilot`. In Chapter 5, we evaluate how SRD and ASR can be optimally combined to produce SRD-tagged transcripts.

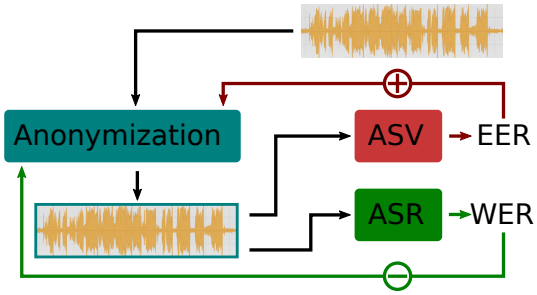


Figure 2.12: Speaker anonymization pipeline. The anonymization network is tuned to reduce the WER reached by an ASR system on the anonymized speech and simultaneously increase the EER of the ASV system.

2.4.4 Speaker Anonymization

The training of neural networks requires large amounts of data. Especially when large amounts of data are scraped from the internet, for example for training LLMs, or data is sent to the cloud for processing, as it is done in ChatGPT, data protection and **privacy** concerns arise (Sebastian, 2023). In Europe, there are two main legislative texts that regulate AI, the general data protection regulation (**GDPR**) (Aridor et al., 2020) and the **EU AI act** (Neuwirth, 2022). These legislative texts forbid, among others, to use biometric data for model training without a permit. Biometric data is data that can be used to unambiguously identify a person. This includes for example voice and image samples of a person. In case of voice data, **speaker anonymization** is an effective method to anonymize the data, which renders it useless for biometric identification, but still allows it to be used for training ML systems (Yoo et al., 2020). This is done by altering speaker-unique features, such as pitch of the speech. In x-vector anonymization, the extracted speaker embedding (**x-vector**) is altered to achieve anonymization (Srivastava et al., 2022). Figure 2.12 visualizes a general speaker anonymization pipeline. In most cases, the objective of a speaker anonymization training is to decrease the WER reached by an ASR system on the anonymized data. Simultaneously the **equal error rate** (**EER**) reached by an **automatic speaker verification**

system (**ASV**) on the anonymized data should be increased. The equal error rate is defined by

$$EER = FAR + FRR, \tag{2.14}$$

with the false acceptance rate (FAR) and the false rejection rate (FRR) (Franzreb et al., 2023). A better overview of the anonymization task and different speaker anonymization networks is beyond the scope of this chapter and can be found in the description of the VoicePrivacy Challenge 2024 (Tomashenko, Srivastava, et al., 2022).

Data protection is also a big hindrance, when it comes to collecting training data from air navigation service providers (ANSPs). In Chapter 3, we investigate therefore, if anonymized ATCO speech can be used for stress detection.

2.4.5 Low-resource learning

The basis of supervised learning is the availability of labeled data. If labeled data is not available, the data must be labeled manually. This labeling process can be very time and cost intensive (Fredriksson et al., 2020). To overcome this, there are multiple approaches. Approaches that focus on generating more training data are for example classic **data augmentation**, **active learning** and **noisy labeling**. A standard data augmentation technique is to apply noise (Mumuni et al., 2022). In case of text data, LLMs can be utilized to produce similar samples (Ding et al., 2024). In contrast to data augmentation, noisy labeling utilizes unlabeled data (Song et al., 2023). The idea behind this is, that even if there is just a small amount of labeled data available, there exists usually a high amount of unlabeled data. To utilize this data, rule-based labeling can be applied. *"Label every sentence with the word* `happy` *in it as* `positive emotion`*"* would be one simple rule. This rule would for example produce a wrong label for *"I am not happy"*. Therefore, the labels produced by such a rule-based system are noisy. The first networks that trained on noisy labels relied on noise adaptation layers to counteract the noise. It has however been shown, that transformer-based architectures do not need these

measures (D. Zhu et al., 2022). After training on noisy labeled data, a model can be further finetuned on clean data.

Active learning is a technique, that is not based on producing a high amount of samples, like the aforementioned methods, but on selecting the best samples for labeling (Ren et al., 2022). To achieve this, a model is trained on an initial pool of labeled data. This model is then used to predict the label of a small part of the unlabeled data. Good candidates for further labeling are the samples, at which the model has the highest uncertainty. Those samples have a high chance to lie, in a classification problem, close to the decision boundary between two classes. They are therefore the samples that have the highest information content. After labeling those samples, the model can be trained again on the now bigger labeled pool.

If there is just little data available, class imbalance often becomes a common problem. Class imbalance in a classification scenario leads to overfitting on the majority class. This can be avoided by data distribution or loss function adjustment. **Under-sampling** the majority class or **over-sampling** the minority class leads to more equally distributed classes and therefore avoids overfitting (Johnson et al., 2019). Introducing class-specific weights in the loss function is another way to avoid overfitting on specific classes. An example of this is the **weighted cross-entropy function** (Aurelio et al., 2019).

In Chapter 6, we use data augmentation to enhance call-sign recognition. In Chapter 8, we investigate the usage of data augmentation, noisy labeling and loss function weighting in read-back error detection.

3

Less Stress, More Privacy: Stress Detection on Anonymized Speech of Air-traffic Controllers

Contents

Air-traffic control (ATC) demands multitasking under time pressure with high consequences of an error. This can induce stress. Detecting stress is a key

point in maintaining the high safety standards of ATC. However, processing ATC voice data entails privacy restrictions, e.g. the General Data Protection Regulation (GDPR) law. Anonymizing the ATC voice data is one way to comply with these restrictions. In this chapter, different architectures for stress detection for anonymized ATCO speech are evaluated. Our best networks reach a stress detection accuracy of 93.6% on an anonymized version of the Speech Under Simulated and Actual Stress (SUSAS) dataset and an accuracy of 80.1% on our anonymized ATC simulation dataset. This shows that privacy does not have to be an impediment in building well-performing deep learning-based models.

The content of this chapter is based on:

> Viswanathan, Janaki, **Blatt, Alexander**, Konrad Hagemann, and Dietrich Klakow (Dec. 2022). "Less Stress, More Privacy: Stress Detection on Anonymized Speech of Air Traffic Controllers." In: *Innovation im Fokus* 2, pp. 43–50. URL: https://www.dfs.de/homepage/de/medien/publikationen/internet-fokus2202.pdf?cid=hrf.

3.1 Introduction

Air-traffic controllers (ATCOs) constantly deal with a lot of information and need to choose the right procedure based on the circumstances and make quick decisions. The high level of responsibility along with the potentially fatal consequences of an error and working in shifts are known as prime sources of occupational stress (Costa, 1996). Measures taken to prevent burn-outs and ATC-related incidents (Nikšić et al., 2022) include mandatory recovery breaks and continuous training of the ATCOs to handle stress and infrequent scenarios (Costa, 1996). However, people cope with stress differently, which includes the behaviour during stress as well as the recovery time needed after stress. ATCO stress detection is an

effective way to prevent incidents (Loewenthal et al., 2000). Monitoring ATCOs' mental state can be done in several ways. One approach is to use physiological measures like heart rate or respiration rate (Sammito et al., 2016). This has the drawback that these methods are intrusive and therefore not suitable for daily use in ATC. A less intrusive approach is to use operational speech data that is recorded anyway and is regularly deleted. Although stress detection for ATC speech is complicated by the fact that ATCOs are trained to remain calm even in stressful situations, Luig et al. (Luig et al., 2010) have already shown with simulated data that speech can be used to measure the workload of an ATCO. In their work, the authors argue that "stress" can be used as a term that describes "an individual's subjective capacity [...] influenced by a multitude of factors" such as working conditions as well as "remarkable events and changes in private life" (Luig et al., 2010). Single influences on this mental state are regarded as "stressors". There exist several works which describe stress as a factor that affects workload (Costa, 1996; Sillard et al., 2000; Hagmüller et al., 2006; John H.L. Hansen et al., 2007; Luig et al., 2010). According to Luig et al., the workload level is describing the subjective capacity utilization, which cannot be directly derived from the taskload level (related to the task complexity or size, e.g. traffic type or amount of traffic). They are targeting the development of a speech analysis system for ATCO voice that indicates different factors of human stress with the goal to estimate the ATCOs workload level from the stress level. In contrary to that, in this work, we use subjective ISA workload measurements to estimate levels, which are then used for ATC speech-based binary stress classification. A major restriction for any ATCO monitoring activities is privacy laws and regulations. Since ATC is a worldwide business, global and also local privacy laws must be met. With the rising collection of speech-assisted tools, there are also new guidelines that have to be met (Politou et al., 2018). One way to avoid privacy- related issues is to remove personal information from the collected data. This can be done either on a text or speech level. On the text level, entities which are linked to private information, for example, birth dates or phone numbers, can be masked or replaced (Adelani et al.,

2020). Since ATC speech is standardized and relies on a fixed phraseology[1], private entities are not as common as in normal speech. Therefore, we focus on speech in this work. On the speech level, anonymization assures that the original speakers - ATCOs or pilots, cannot be tracked back (Tomashenko, X. Wang, et al., 2022). In the scope of this work, therefore, a stress recognition model for anonymized ATCO speech is proposed. In addition, a multiclass speaking style classification task is implemented to show that privacy does not have to be a barrier for speech processing.

3.2 Related work

Traditional speech-based stress or emotion identifying methods are rule-based or use Hidden Markov Models (HMMs) (Nogueiras et al., 2001). More recent approaches rely on deep learning methods. Tomba et al. (Tomba et al., 2018) show that mean energy, mean intensity and mel frequency cepstral coefficients (MFCC) can be used to detect stress. Luig et al. (Luig et al., 2010) investigate different speech features for ATCO workload prediction. They use the frequency of utterances spoken per minute as an indirect indicator of stress. Borghini et al. propose to measure ATCO stress directly from brain activities using methods such as electroencephalography (EEG) (Borghini et al., 2020). In (Shin et al., 2020), the authors propose different model architectures based on deep learning algorithms. They use convolutional layers to embed the relevant spectral input features and propose to add a long short-term memory (LSTM) network on top of the convolutional layers to capture the temporal components. The final multi-head attention layer can give more weight to the important parts of the input. This design is taken as the basis for our stress recognition model. Xu et al. (Xu et al., 2021) propose a similar architecture for emotion recognition and identify vocal tract length perturbation (VTLP) as a useful augmentation method for emotion

1 ATC phraseology examples from the Federal Aviation Administration: https://www.faa.gov/air_traffic/publications/atpubs/aim_html/chap4_section_2.html

recognition. Speaker anonymization methods are benchmarked since 2020 in the Voice Privacy Challenge (VPC) (Tomashenko, X. Wang, et al., 2022). For privacy evaluation, the VPC2020 considers various attack scenarios depending on the knowledge of the attacker. In the first task, *unprotected*, both the users and the attackers use original data. In the second task, *ignorant attacker*, users anonymize their data but the attackers are unaware of it and assume original data. In the third task, *lazy-informed*,both, the users and attackers, use anonymized data and the attacker also has access to the speaker identities. For the work at hand, the speaker anonymization method of Kai et al. (Kai et al., 2021) was used since it reaches equal error rates (EERs) above 40% on task II of the VPC2020 which indicates a high anonymization capability. The automatic speech recognition (ASR) method of the VPC2020 reaches a low word error rate (WER) of up to 10% on the anonymized speech, which indicates that the anonymization of Kai et al. still allows the recognition of the spoken words. However, other downstream tasks, i.e. applications of anonymized speech, are not investigated in the VPC2020. Therefore, an evaluation of emotion recognition has been included in this work.

3.3 Experimental Setup

3.3.1 Datasets

Our experiments are performed on the SUSAS (John HL Hansen et al., 1997) and DFS Munich approach simulation (DFS-MAS) datasets. The SUSAS dataset contains speech samples for different speaking styles. Nine speaking styles are considered: *anger*, *fast*, *Lombard* (involuntarily increase of the voice level when there is background noise (Zollinger et al., 2011)), *loud*, *clear*, *neutral*, *slow*, *soft* and *question*. Each speaking stile has 630 samples each except for *neutral* with 631 samples. Hence, the considered SUSAS dataset consists of 5671 samples in

total. To enable binary stress detection, the following grouping of the speaking styles is suggested:

- STRESS: *anger*, *fast*, *Lombard*, *loud*
- NO-STRESS: *clear*, *neutral*, *slow*, *soft*

The label *question* has been left out for our binary classification since it could occur in both stress and no-stress scenarios. Hence, there are 5041 samples for the stress detection task. An 80:20 split is done to create train and test sets and the train set is split again as 80:20 to create train and validation datasets. This is based on the approach used by Shin et al. (Shin et al., 2020) and results in a train |val |test split of 64% |16% |20%. This data split is used for all experiments. The DFS-MAS dataset was produced by Deutsche Flugsicherung GmbH (DFS). It consists of ATC simulation data for Munich approach. Two male and two female ATCOs, each with more than ten years of work experience, have been recorded for this dataset. Following the approach by Luig et al. (Luig et al., 2010) described above, the workload level is used here as an approximation for the stress level of an ATCO. During the 90-minute simulation run, the workload of the ATCOs was measured every five minutes via an electronically presented pop-up questionnaire using the instantaneous self- assessment of workload technique (ISA) (Jordan et al., 1992; Kirwan et al., 1997). For binary stress detection on the DFS-MAS dataset, the stress labels are grouped according to the ISA workload labels:

- STRESS: *high*, *excessive*
- NO-STRESS: *boring*, *relaxed*, *comfortable*

The DFS-MAS data is highly imbalanced with 60 stress and 678 no-stress samples. Therefore, data augmentation methods such as VTLP and white noise addition are applied. To ensure that the distribution of the augmented data is the same across labels, the same number of augmented samples are generated for both classes. The standard parameters in the *nlpaug* package (Ma et al., 2020) are used. We generate ten different augmented versions of the stress samples - five

Table 3.1: Summary of the augmented DFS-MAS dataset. The multiplication factors of the [train, validation, test] splits represent the number of different copies created per clean sample.

Augmentation method	Stress [39, 9, 12]	No-stress [435, 108, 135]
None	[39, 9, 12]	[45, 18, 15]
VTLP	[39, 9, 12] * 5	[195, 45, 60] * 1
White noise	[39, 9, 12] * 5	[195, 45, 60] * 1
Total	[429, 99, 132]	[435, 108, 135]

using VTLP and five using white noise addition, while the majority class (No-Stress) is just augmented once per sample. Table 3.1 gives an overview of the data augmentation. To test the performance of stress detection on anonymized data, an anonymized version of both datasets is created. The anonymization method is described in the next section. The classification tasks are performed on both, anonymized and non- anonymized data.

3.3.2 Anonymization

As mentioned above, the lightweight voice anonymization (LVA) of Kai et al. (Kai et al., 2021) is used as the speaker anonymization method. Due to the high overall performance in the VPC2020 Tasks I, III, and V, waveform resampling is used as the anonymization method for the experiments if not stated otherwise. Moreover, the gender-specific parameters are used for all samples. Resampling is based on the Waveform Similarity Overlap-Add (WSOLA) algorithm, which allows stretching the original speech signal by a factor , while maintaining the correct pitch. Resampling this stretched signal by an -times faster sampling frequency

leads to the anonymized signal, which is of equal length as the original signal but varies, for example, in the pitch and formants.

3.3.3 Speech Preprocessing

The ATC utterances are pre-processed before they are fed through the classification network. A Wiener filter (Benesty et al., 2005) is first applied to remove noise. Furthermore, a pre-emphasis filter is applied which boosts the signal-to-noise ratio of the higher frequency components since they are more susceptible to noise. Short-time Fourier transformation (STFT) is applied to generate the spectrogram. Then the log-amplitude spectrogram is obtained by taking the logarithm of the amplitude component of the spectrogram. It is further converted to a mel spectrogram (MS) using the mel frequency conversion formula (Stevens et al., 1937) together with a filter bank of 128 filters. Two different speech representations are compared. The first one is obtained by applying the logarithm to the MS. This results in the log mel spectrogram (LMS) as network input. The second speech variant is generated by applying the discrete cosine transformation (DCT) to the LMS to generate the mel frequency cepstral coefficients (MFCC) (Chadawan Ittichaichareon et al., 2012). Using MFCC has the advantage that the input data can be compressed without losing too much information by using the most informative DCT coefficients and dropping the rest. For our experiments, 20 coefficients are used.

3.3.4 Stress Detection Networks

Our stress detection networks are based on (Shin et al., 2020). Three different architectures are investigated with increasing complexity - CNN, CRNN, and CRNN+Attention. They are built using different parts of the stack: CNN + LSTM + multi-head attention. Figure 3.1 shows the architecture of the models.

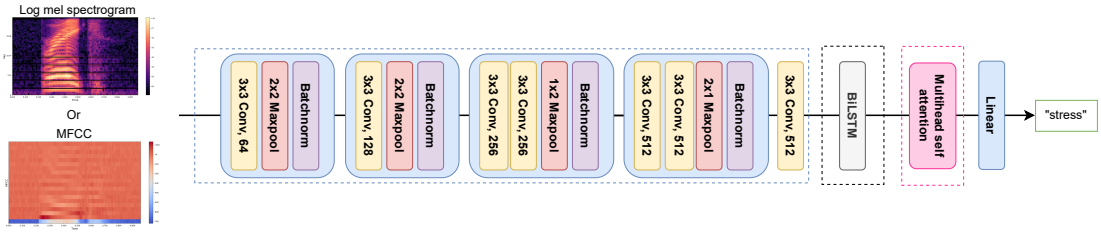


Figure 3.1: Stress detection network depicting all three architectures. The network is built incrementally. The blue dotted box represents the CNN, CNN along with the black dotted box represents the CRNN, and the CRNN along with the pink dotted box represents the CRNN+Attention model architecture.

Table 3.2: Comparison of architecture sizes for different speech representations.

Model architecture	Number of trainable parameters	
	MFCC	LMS
CNN	7,435,906	8,114,818
CRNN	9,012,866	9,691,778
CRNN+Attention	10,063,490	10,742,402

Multi-head attention with four heads is used since this is the best performing architecture of Shin et al. (Shin et al., 2020). The experiments are repeated thrice and the mean and the standard deviation of the accuracies are calculated to check for robustness of the models.

3.4 Results

3.4.1 Architecture Comparison

The different architectures vary largely in their number of trainable parameters as shown in Table 3.2. This raises the question whether the additional parameters

Table 3.3: Emotion and stress recognition accuracies on the SUSAS and DFS-MAS test sets. The standard deviation scores are given in brackets.

			SUSAS		DFS-MAS
Anonymized	Feature	Model architecture	9 Emotions	Stress	Stress
No	MFCC	CNN	75.6% [0.008]	93.0% [0.009]	74.2% [0.013]
		CRNN	**75.8%** [0.008]	93.1% [0.006]	73.5% [0.012]
		CRNN+Attention	70.6% [0.029]	**93.9%** [0.006]	**75.6%** [0.039]
	Log mel spectrogram	CNN	76.8% [0.004]	93.6% [0.002]	66.9% [0.027]
		CRNN	**77.7%** [0.006]	**94.4%** [0.004]	66.9% [0.054]
		CRNN+Attention	73.9% [0.022]	93.0% [0.005]	**71.6%** [0.042]
Yes	MFCC	CNN	**73.7%** [0.006]	91.2% [0.002]	71.8% [0.042]
		CRNN	72.3% [0.008]	91.5% [0.005]	69.5% [0.046]
		CRNN+Attention	71.5% [0.009]	**91.9%** [0.004]	**75.9%** [0.044]
	Log mel spectrogram	CNN	74.9% [0.008]	92.5% [0.005]	71.4% [0.006]
		CRNN	**75.6%** [0.015]	**93.6%** [0.003]	74.8% [0.036]
		CRNN+Attention	74.1% [0.003]	**93.6%** [0.002]	**80.1%**[3.384]

lead to an increased accuracy. The architecture comparison in Table 3.3 shows that either the CRNN or CRNN+Attention models have the highest accuracy for most of the experiments. The highest scores on the speaking style and stress classification tasks on the SUSAS dataset are reached by the CRNN architecture in combination with the LMS feature. This holds true for anonymized and non-anonymized data where the CRNN model outperforms the 11% larger CRNN+Attention model. In contrast, on the DFS-MAS dataset, the benefit of the additional attention layer of the CRNN+Attention model leads to a significant increase in accuracy of more than 5% in comparison with the CRNN model.

Replacing MFCC with LMS as input feature leads to an average performance gain of 1-2%. This comes with the trade-off that the input dimension is increased by a factor of 6.4. Therefore, using MFCC as input is a valid alternative for devices with lower computational power.

3.4.2 Stress Detection for ATC

ATC speech differs substantially from normal speech. It consists of a set of phraseologies that allow for handling different situations, such as landing, take-off, and emergencies. In addition to that, ATCOs are supposed to give clear and calm instructions even under stressful situations. Furthermore, it is difficult to obtain a properly labeled, well-balanced dataset specific to the ATC scenario. This makes stress detection in this domain challenging. Table 3.3 shows the difference in the accuracy of stress detection between the SUSAS and the DFS-MAS dataset. Due to the challenges mentioned above, the mean accuracy on the DFS-MAS dataset is about 20% lower. In contrast to the SUSAS data, the more complex CRNN+Attention model reaches the highest accuracies independent of the input features and the anonymization. This is another indicator of the difficulty level of the DFS-MAS dataset. However, our best model reaches a performance of 80.1% on the DFS-MAS dataset.

3.4.3 Anonymization Impact

Table 3.3 allows the comparison of the model performance of anonymized and non-anonymized datasets trained using different model architectures and different speech features. On the SUSAS dataset, the models trained on the anonymized version have a mean average accuracy that is 1-2% less than its non- anonymized counterpart. For the CRNN+Attention network, anonymization even leads to a performance increase. Figure 3.2 gives a more detailed insight into the classification accuracy for each class. For both anonymized and non-anonymized data, the CRNN model with LMS feature classifies the majority of the classes correctly, with an accuracy of over 70%. On both datasets, the model has problems distinguishing similar classes, such as *clear*, *neutral*, and *slow* from another. The majority of the performance drop from non-anonymized to anonymized data is due to the

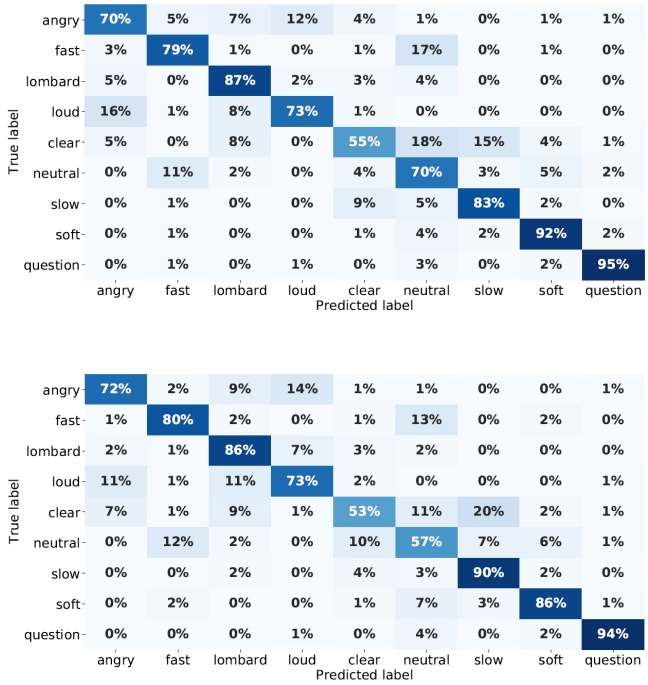


Figure 3.2: Confusion matrices of the CRNN model with LMS as feature on the non-anonymized (at the top) and anonymized (at the bottom) SUSAS dataset.

misclassification of *neutral* speech, where the accuracy drops from 70% to 57%. For the other classes, the accuracy difference is 7% or less.

The anonymization method of the target data for stress detection might not always be known. Therefore, the question is, how would the performance decrease if the inference data is anonymized while the model is trained on non- anonymized data. Comparing the first row of Table 3.4 with the best-performing models in Table 3.3 shows that the model performance decreases substantially for both model architectures. While the accuracy of the CRNN+Attention model using MFCC drops from 93.9% to 80.1%, the accuracy of the CRNN model using LMS features has an almost 19% drop - from 94.4% to 75.7%. The results are similar when the model is trained on raw SUSAS data and tested on anonymized SUSAS data.

On the ATC-relevant DFS-MAS dataset, the anonymization leads to an increase in performance. The best performing network, CRNN+Attention, trained and tested on anonymized data outperforms the best model for non-anonymized

Table 3.4: Stress recognition cross-domain test accuracies. The best performing models of Table 3.3 are used for testing. *(A)* represents the corresponding anonymized dataset.

Trained on	Tested on	MFCC	LMS
SUSAS	SUSAS **(A)**	80.1%	75.7%
		[ATTN]	*[CRNN]*
SUSAS **(A)**	SUSAS	80.6%	78.7%
		[ATTN]	*[ATTN]*
SUSAS	DFS-MAS	50.2%	50.2%
		[ATTN]	*[CRNN]*
SUSAS **(A)**	DFS-MAS	64.8%	51.3%
		[ATTN]	*[ATTN]*
SUSAS	DFS-MAS **(A)**	56.2%	45.3%
		[ATTN]	*[CRNN]*
SUSAS **(A)**	DFS-MAS **(A)**	72.3%	52.1%
		[ATTN]	*[ATTN]*

data by 4.5%. Since the non-augmented DFS-MAS dataset is imbalanced, with less than 100 stress utterances, the anonymization could act as an additional augmentation method. It should be noted that the CNN and CRNN models do not benefit from the anonymization, but they are also outperformed by the attention model by 1.4% to 9.7%.

3.4.4 Cross-Domain Stress Detection

To the best of our knowledge, there are no publicly available stress-labeled ATC datasets. Therefore, it is also evaluated if it is possible to reach high stress recognition results on ATC data with a model that is trained on another domain. The results are shown in Table 3.4. For this, the best-performing SUSAS models,

as marked in bold in Table 3.3, are used on the out-of-domain ATC data. In contrast to the results in Table 3.3, anonymizing the SUSAS dataset improves the cross-domain performance significantly by over 14% for the CRNN+Attention model with MFCC input features. The additional augmented data counteracts domain overfitting and leads therefore to a better generalization of the model. By adding anonymization also to the DFS-MAS test set, the performance increases over 22% compared to the nonanonymized datasets. With an accuracy of 72.3%, the difference to the best- performing model trained on the ATC data is below 8%. Interestingly, using MFCC as input gives consistently better cross-domain scores than using LMS as input. The higher information condensation in MFCC leads to a better generalization and hence avoids overfitting to the training domain, similar to anonymization.

3.5 Conclusion

Our experiments show that anonymization is not an obstacle to stress and speaking style recognition. In fact, it is observed that anonymization causes just a minor accuracy drop of 1-2% on the SUSAS dataset and even leads to a performance increase on the target ATCO speech of more than 4%. This probably comes down to the fact that anonymization can be seen as a data augmentation method, which could be beneficial, especially for low-resource tasks. Furthermore, we see that on the single speech style level, the performance drop is mainly due to the misclassification of *neutral* speech samples with, for example, similar *clear* speech samples. In other words, the classification results are stable through anonymization in the majority of the classes. In the cross-domain setting, it is shown that stress recognition models trained on out-of-domain data can be used to perform stress prediction on ATC. In this case, one should rely on MFCC as input since they generalize better than the LMS input. For our anonymization method, it is shown that if the anonymization method for ATC data is known, anonymizing the out-of-domain training data additionally improves the performance. Regarding

the architectures, it is shown that a combination of MFCC input and the CRNN model outperforms the CRNN+Attention models using the LMS feature in the speaking style recognition task, while having only 84% of its trainable parameters. This makes this model interesting if computational power is a limiting factor. Nevertheless, on the more demanding ATC data, the CRNN+Attention architecture outperforms the other networks by a margin, this holds also true for the cross-domain experiments.

For future work, we would like to explore different data augmentation methods that might increase accuracy. Furthermore, we would like to investigate MFCC with a different number of coefficients as an input feature since we observed equally good results as LMS. Another aspect to explore is transfer learning, since it has proved to be as good as the trained models on the DFS-MAS dataset. With transfer learning and the comparatively lower dimensional MFCC as an input feature, we could expand our work to have more practical applications where we could reduce the space and computational complexity to get live predictions and also train on edge devices. By having a live stress detector, we could actively reduce the workload stress of ATCOs and avoid any incidents.

In summary, it is strongly suggested to test the incorporation of anonymization methods for privacy- critical tasks, especially for air traffic control.

4

Acoustic and Lexical Adaptation of Wav2vec 2.0: a Case Study in Air-traffic Control

Contents

Transformer neural networks have shown remarkable success on standard automatic speech recognition (ASR) benchmarks. However, they are known to be less robust against domain mismatch, particularly with air-traffic control (ATC) speech data. In the ATC domain, transformer-based ASR systems generally do not transfer across different datasets. The reasons for the poor transferability between ATC datasets remain unclear. In this chapter, we therefore investigate the influence of acoustic variability and lexical differences on the ASR perfor-

mance across various ATC datasets. By fine-tuning and evaluating wav2vec 2.0 on synthetic ATC datasets, we examine the effect of acoustic variability on the model performance. Furthermore, we assess the effect of lexical differences by correlating language model perplexity with performance. Our findings reveal that a combination of acoustic and lexical mismatch causes the bad inter-dataset transferability and give insights on how to improve future ASR models for ATC.

The content of this chapter is based on:

> **Blatt, Alexander**, Badr M. Abdullah, and Dietrich Klakow (2023). "Ending the Blind Flight: Analyzing the Impact of Acoustic and Lexical Factors on WAV2VEC 2.0 in Air-Traffic Control." In: *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1–8. DOI: 10.1109/ASRU57964.2023.10389646.

4.1 Introduction

Automatic speech recognition (ASR) is the first step in a speech-processing pipeline for air-traffic control (ATC) communication. ATC communication consists of instructions from an air-traffic controller (ATCO) to a specific pilot and a read-back from that pilot [1]. In recent years, several corpora for ATC-ASR have been gathered (Zuluaga-Gomez, Motlicek, et al., 2020). However, apart from the ATCOSIM corpus (Hofbauer et al., 2008) and a one hour chunk of the ATCO2 corpus (Zuluaga-Gomez, Veselý, Szöke, et al., 2022), datasets are either not available without a fee or not publicly available at all. Since ATC communication is formalized and has a unique phraseology (Helmke, Slotty, et al., 2018), out-of-domain (OOD) trained ASR models transfer poorly to ATC data (Zuluaga-Gomez, Prasad, Nigmatulina, S. S. Sarfjoo, et al., 2023b; Krishnan et al., 2023). Despite

1 Communication examples: https://wiki.flightgear.org/ATC_phraseology

these challenges, some ATC-ASR models have been developed in recent years. While earlier models rely on Kaldi (Kocour, Veselý, Blatt, et al., 2021), newer approaches are based on pretrained transformer models such as wav2vec 2.0 (Zuluaga-Gomez, Prasad, Nigmatulina, S. S. Sarfjoo, et al., 2023b). Although those models are built on several hours of training data that even incorporate non-publicly available data, high word error rate (WER) variations in-between different ASR benchmark corpora have been observed (Kocour, Veselý, Blatt, et al., 2021; Zuluaga-Gomez, Motlicek, et al., 2020; Zuluaga-Gomez, Prasad, Nigmatulina, S. S. Sarfjoo, et al., 2023b). Previous works on ATC-ASR focused therefore on using newer or more parameter-rich models to increase the overall performance on the benchmark datasets. In contrast to these works, we will explore the causes of poor transferability across the different ATC datasets at the example of wav2vec 2.0. This will not only give a better understanding on how to interpret the WERs reached on the individual benchmark datasets, but also allow one to develop better ATC-ASR models in the future. In the following sections, we analyze the influence of the acoustic variability. Furthermore, we model the acoustic variability by adding Gaussian noise of different levels to text-to-speech (TTS) generated versions of the datasets. Regarding lexical differences, we analyze intra and cross-dataset perplexities and out-of-vocabulary (OOV) rates. To gain a better understanding of the wav2vec 2.0 adaptation to the ATC corpora, we additionally analyze the internal changes of the wav2vec 2.0 architecture during fine-tuning on the different ATC corpora. In the next section, we will elaborate related studies in the fields of ATC and explainability of transformer-based ASR.

4.2 Related Work

Zuluaga-Gomez et al. have trained wav2vec 2.0 and XLS-R for ATC speech recognition and provide results over different ATC datasets (Zuluaga-Gomez, Prasad, Nigmatulina, S. S. Sarfjoo, et al., 2023b), the resulting WERs of their best model still show a significant variation over the test datasets. One way to

make wav2vec 2.0 more robust has been introduced by Zhu et al. (Q. S. Zhu et al., 2022). They force the feature encoder to generate speech representations for a noisy speech input that resemble representations for clean speech. The resulting model has a superior noise tolerance in comparison to the baseline wav2vec 2.0 model. This shows, on the other hand, the sensitivity of the standard transformer-based ASR models to noise. Hu et al. (Hu et al., 2023) have built on this work to develop a wav2vec 2.0 based model that does speech enhancement without introducing artifacts that deteriorate the ASR performance. A method to deal with the low availability of labeled in-domain data has been proposed by Hsu et al. (Hsu et al., 2021). They have shown that if there is no in-domain data available for finetuning, using unlabeled in-domain data during pretraining can give a significant performance improvement. For our wav2vec 2.0 feature analysis, we build on the following two previous works. Phang et al. (Phang et al., 2021) have shown that the centered kernel alignment (CKA) similarity scores of text-based transformer models show same-similarity clusters along the diagonal after they are fine-tuned. Choi et al. (Choi et al., 2022) have shown that the information encoded in a wav2vec feature encoder is analogous to a spectrogram and that closer latent representations imply acoustic similarity.

4.3 Experimental Setup

The ATC **datasets** used in the following experiments are listed in Table 4.1. The ATCOSIM corpus (Hofbauer et al., 2008) consists of simulated conversations between air-traffic controllers and pilots. Since the recordings were made in a controlled environment, the speech is less noisy than for the following two corpora. The ATCO2 corpus (Zuluaga-Gomez, Veselý, Szöke, et al., 2022) contains real ATC conversations from various, mostly European airports and was recorded during the ATCO2 project with VHF-receivers [2]. The LiveATC corpus consists of

2 Receiver guide: https://ui.atc.opensky-network.org/intro

Table 4.1: Dataset splits used for the experiments. The mean utterance length for each dataset is roughly four seconds. In the last column, the mean SNR over the full dataset is given.

Dataset	Train	Val	Test	SNR
ATCO2	2739	342	343	13.1
ATCOSIM	2286	286	286	29.4
LiveATC	512	-	518	7.2

two subcorpora, LiveATC1 and LiveATC2 (Zuluaga-Gomez, Veselý, Blatt, et al., 2020a), both gathered during the ATCO2 project from the LiveATC web-page [3], a web-page broadcasting live ATC conversations.

Fine-tuning wav2vec 2.0 on ATC data is done by training `wav2vec2-base` [4] for 40 epochs on the train-split of the datasets in Table 4.1. After fine-tuning, the checkpoint model with the lowest WER score on the validation set is used for testing.

To generate **text-to-speech (TTS)** versions of the aforementioned datasets out of the transcripts, we use the VITS model (Variational Inference with adversarial learning for end-to-end Text-to-Speech) (J. Kim et al., 2021) from the Coqui-AI library [5]. The model can be described as a conditional variational autoencoder and produces natural sounding speech from text. The male speaker 226 is chosen from the list of speakers, as it produces the most realistic ATC speech. To generate our synthetic noisy ATC data, we add Gaussian noise to the TTS versions of the datasets.

To overcome the problem of missing clean versions of the ATC datasets to calculate the **signal-to-noise ratio** (SNR), we use the WADA-SNR approach introduced by Kim et al. (C. Kim et al., 2008) to obtain a robust estimate for the SNR. To ensure consistency, all SNR values mentioned in this work are based on

3 LiveATC webpage: https://www.liveatc.net/
4 Wav2vec 2.0 model: https://huggingface.co/facebook/wav2vec2-base
5 Coqui-AI webpage: https://github.com/coqui-ai/TTS

Table 4.2: Word and character error rates across the different ATC datasets depending on the training set. All scores are generated by finetuning and testing *wav2vec2-base* on the datasets, except for the last row, where *wav2vec2-base-960h* is used, which is already finetuned on LibriSpeech. Intra-dataset scores are marked blue.

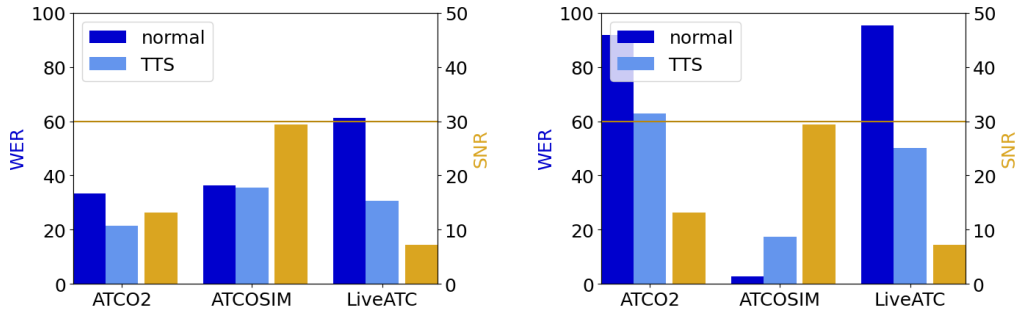
Training Data	ATCO2		ATCOSIM		LiveATC	
	WER (%)	CER (%)	WER (%)	CER (%)	WER (%)	CER (%)
ATCO2	33.4	20.4	36.6	16.8	61.2	40.3
ATCOSIM	91.9	61.5	2.67	1.00	101.9	67.8
LibriSpeech	99.6	64.6	71.0	32.0	103.4	70.5

this method. Experimental validations on the synthetic noisy ATC datasets have shown that the WADA-SNR scores show just small deviations from the actual SNR values.

To measure the wav2vec 2.0 **feature similarities**, when fine-tuned on different datasets, we apply the centered kernel alignment (CKA) method as similarity measure, since it is well defined for small sample sizes, in contrast to other similarity measures like CCA and pwCCA (Kornblith et al., 2019). The output-layer features of the convolutional blocks of the feature encoder and the dense-layer features of the transformer encoder are mean-pooled over the sentence length before comparison.

4.4 Results

As already observed in previous works (Kocour, Veselý, Blatt, et al., 2021; Zuluaga-Gomez, Motlicek, et al., 2020; Zuluaga-Gomez, Prasad, Nigmatulina, S. S. Sarfjoo, et al., 2023b), the performance of an ASR model varies depending on the target and training dataset. Even if all datasets come from the same domain, namely



(a) ATCO2 (b) ATCOSIM

Figure 4.1: WER on the standard and TTS versions of the ATC datasets. All scores are generated by fine-tuning and testing *wav2vec2-base* on ATCO2 (a) and ATCOSIM (b) data. The border to clean speech SNR>30 is marked (Grimaldi et al., 2018).

air-traffic control, the word error rate and character error rate (CER) vary, as Table 4.2 shows.

However, WER and CER correlate across all datasets and there are no dataset-specific WER/CER ratios. Without including intra-dataset scores, the lowest WER/CER ratios are reached on ATCOSIM followed by ATCO2 and LiveATC. This correlates inversely with the SNR values given in Table 4.1. The last WER column of Table 4.2 shows the importance of in-domain fine-tuning. Wav2vec 2.0 fine-tuned on ATCO2 reaches a WER 40-50% lower than the model fine-tuned on the OOD LibriSpeech corpus. Surprisingly, if wav2vec 2.0 is fine-tuned on ATCOSIM, this difference is much smaller. In the following, we will evaluate this and analyze which acoustic and lexical differences exist between the datasets and how wav2vec 2.0 reacts to them.

4.4.1 Acoustic Differences

As discussed above, there seems to be a correlation between the noise level and the word error rate. To rule-out the influence of out-of-vocabulary (OOV) words or other language, respectively lexical-based features, we generate text-to-speech

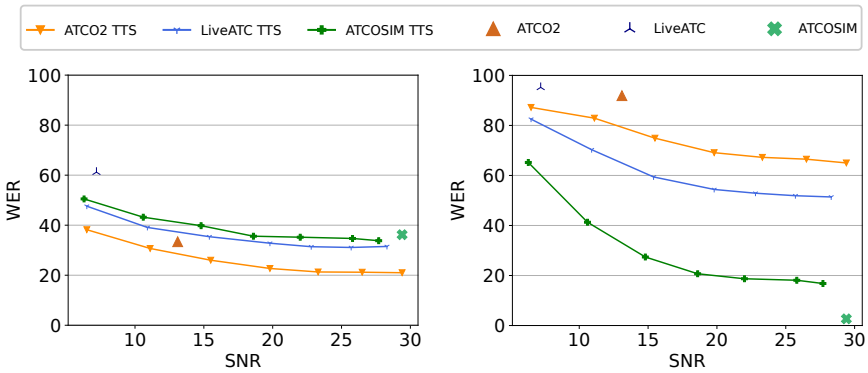
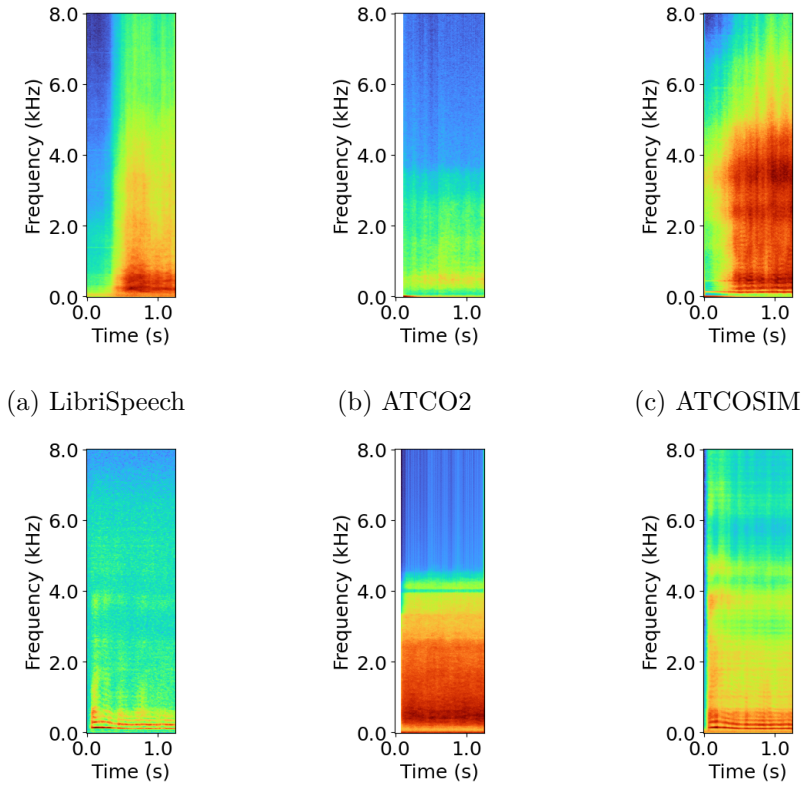


Figure 4.2: WERs on the original and TTS data with Gaussian noise of different levels applied. Wav2vec 2.0 is trained on the original ATCO2 (left) or ATCOSIM (right) dataset.

(TTS) versions of the datasets, as described in Section 4.3, and compare the WERs reached on the datasets. Since they share the same transcripts, all differences between the TTS and non-TTS versions are due to acoustics. Figure 4.1 shows the WERs reached on the TTS and non-TTS versions together with the SNR values of the non-TTS versions taken from Table 4.2. For both training datasets, ATCO2 and ATCOSIM, the difference of the WERs between the TTS and non-TTS versions correlates inversely with the SNR value. This shows that noise is a major cause for the performance degradation of the ASR models on ATC datasets.

In order to evaluate the effect of the noise over a broad range, we add Gaussian noise of different levels to the TTS versions of the datasets. The results are shown together with the WERs reached on the original datasets in Figure 4.2. There are four main observations. Firstly, the higher the noise, respectively, the lower the SNR, the steeper is the gradient of the curves. For SNR levels greater than 25, the effect of the noise is negligible, which is consistent with the definition of clean speech for SNRs>30 of Grimaldi et al. (Grimaldi et al., 2018). The second observation is that the training dataset not only influences the overall WER reached on the test set, but also the noise sensitivity (gradient). The model trained on ATCO2 data (Figure 4.2 left) shows a significantly lower sensitivity to noise than the model trained on less noisy ATCOSIM data (Figure 4.2 right).



(a) LibriSpeech (b) ATCO2 (c) ATCOSIM

(d) LiveATC Gauss (e) LiveATC (f) LiveATC TTS

Figure 4.3: Overlay of spectrograms from 100 samples of the the different ATC datasets and LibriSpeech as reference. The *LiveATC Gauss* spectrogram is based on TTS data with Gaussian noise with an average SNR of 6.5 dB, which is close to the original noise level of LiveATC with 7.2 dB.

The third observation is that for high noise levels with a SNR<10, the model trained on ATCO2 outperforms the ATCOSIM model on the ATCOSIM test data. This indicates that for high-noise target datasets, matching the noise distribution during training can become more important than lexical similarities between the training and test set. The last observation is that wav2vec 2.0 reaches slightly higher WERs on the non-TTS test sets of ATCO2 and LiveATC than on the TTS versions with the same noise level.

To examine this difference further, we overlay the spectrograms of 100 samples from each ATC dataset. To allow an overlay, each recording is trimmed to the same length and the spectrograms are normalized. Figure 4.3 shows the resulting

Table 4.3: Lexical diversity of the ATC datasets, measured with the moving average type-token ratio (MATTR) and the measure of textual lexical diversity (MTLD)

Dataset	MATTR	MTLD
ATCO2	0.635	29.5
ATCOSIM	0.585	26.6
LiveATC	0.581	23.3

spectrograms. The comparison of the datasets shows that each dataset has a unique noise characteristic. In the ATCO2 dataset, the harmonics of the voice stand far less out against the background noise than the harmonics in the ATCOSIM dataset. Furthermore, the LiveATC and ATCO2 dataset spectrograms show a low-pass characteristic, with a loss of signal power over 4 kHz. Additionally, the LiveATC dataset shows a narrow-band signal loss exactly at 4 kHz. The spectrogram of the LiveATC TTS data with Gaussian noise (SNR = 6.5), noticeably differs from the standard LiveATC spectrogram (SNR =7.2). Meaning that the WADA-SNR scores do not reveal the complexity of the noise. This explains why the WER curves on the TTS datasets in Figure 4.2 are lower bounds for the WERs reached on the original datasets. To reproduce the original noise for each dataset, more complex noise types, such as band-pass or low-pass filters, must be included. In the next section, we will evaluate the lexical differences between the datasets.

4.4.2 Lexical differences

We have shown that there exists a correlation between the noise and the WER reached on the datasets. In this section, we will evaluate whether there is a similar correlation for the lexical features. To get a better understanding for the complexity of the datasets, the lexical diversity (LD) is measured via moving average type-token ratio (MATTR) and the measure of textual lexical diversity

Table 4.4: Cross (black) and intra-dataset (blue) perplexities. 4-gram language models are generated for each training dataset.

Training Data	Perplexity on test data		
	ATCO2	ATCOSIM	LiveATC
ATCO2	24.8	138.0	88.2
ATCOSIM	**417.2**	4.8	276.5
LiveATC	144.6	120.4	25.6

(MTLD), which are better estimates for the lexical diversity than other measures, as shown by Tager-Flusberg et al. (Tager-Flusberg, 2015). Table 4.3 shows the diversity scores of the datasets. The LiveATC dataset has the lowest MATTR and MTLD score, indicating that it has the lowest lexical diversity. But the small difference of just 9% to the highest MATTR score, measured on the ATCO2 dataset, shows that the three datasets have a quite similar lexical diversity.

To find more substantial lexical differences, we calculate the cross and intra-dataset perplexities using 4-gram language models (LM). All LMs are generated on the train-splits and tested on the test-splits of the datasets, Table 4.4 shows the results. The highest cross-dataset perplexities are found on the ATCO2 test dataset, indicating the worst transferability of an ASR model trained on the other datasets to this dataset. For the intra-dataset perplexities, the LiveATC and ATCO2 dataset have similar scores, while the ATCOSIM → ATCOSIM perplexity is five times lower. This shows, that the simulated scenarios in ATCOSIM do not have the variability of the operational recordings found in the ATCO2 and LiveATC corpora. This could also be due to the fact that the ATCO2 and LiveATC datasets cover multiple airspaces as stated in Section 4.3. If ATC conversations are recorded in different airspaces for different datasets, this has consequences on the vocabulary. Each airspace has different waypoints, is targeted by different airlines and uses different communication frequencies, to just name a few differences. This also shows in the OOV rates, which can be seen in Table 4.5. The comparison of Table 4.4 and Table 4.5 shows that the perplexities and the OOVs are correlated,

Table 4.5: Cross and intra-dataset (blue) out-of-vocabulary OOV rates in percent.

Training Data	OOV rate on test data (%)		
	ATCO2	ATCOSIM	LiveATC
ATCO2	2.05	7.24	5.20
ATCOSIM	**27.0**	0.65	18.02
LiveATC	11.17	12.90	3.23

with one exception. On the ATCOSIM test data, lower OOV rates are reached with ATCO2 source data, than with LiveATC source data, while it is the other way around for the perplexity. An inspection of the OOVs shows that in the case LiveATC → ATCOSIM, the OOVs contain many German words, like airline names, city names and greetings. These OOVs are missing in the ATCO2 → ATCOSIM case, likely due to the recordings from Swiss airspaces in the ATCO2 dataset.

Since both, perplexity and OOV rates show a lexical mismatch, we want to quantify to which extend this can be fixed by using a 4-gram LM trained on the train-split of the target dataset. Table 4.6 shows the mean results over all source and target dataset combinations, using ATCO2 and ATCOSIM as source data

Table 4.6: Relative WER drop in percent (%), when using a 4-gram LM generated on the train-split of the target dataset. Testing is done on the test-splits of the target datasets. Mean scores over all target-source dataset combinations are given for TTS and non-TTS versions. The absolute difference is given in brackets.

Source Data	rel. WER drop on target data (%)	
	normal	TTS
normal	21.6 (53.42-44.95)	**27.6** (36.4-27.3)
TTS	11.9 (89.01-79.71)	22.8 (19.55-15.58)

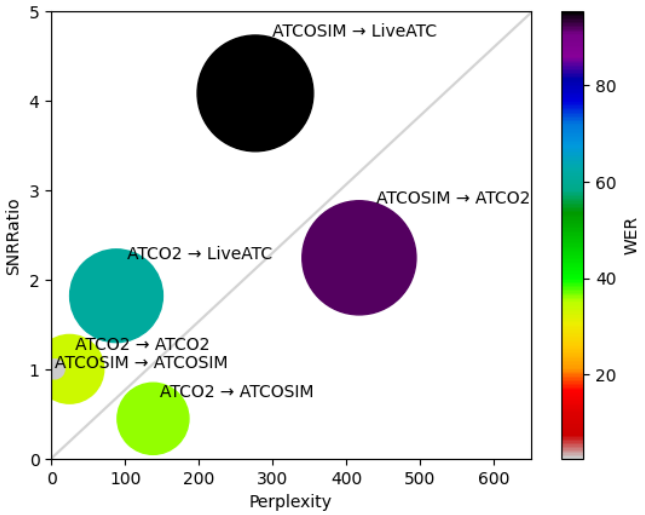


Figure 4.4: WER depending on the relative difference between test and training SNR and the perplexity of a LM generated from training data and evaluated on test data.

and ATCO2, ATCOSIM and LiveATC as target data. For both, source a target data, either TTS or non-TTS versions of the datasets are used.

The resulting scores show that adding the LM on top of wav2vec 2.0 results in the highest improvement for the non-TTS (train) → TTS (test) setting. Interestingly, the relative improvement for TTS → TTS and non-TTS → non-TTS is nearly equivalent. This shows that even if there is an acoustic mismatch, adaptation to the target airspaces via LM can bring a big improvement. In the worst case scenario, TTS → non-TTS, where wav2vec 2.0 has never seen noisy data during training, there is still more than 11% improvement.

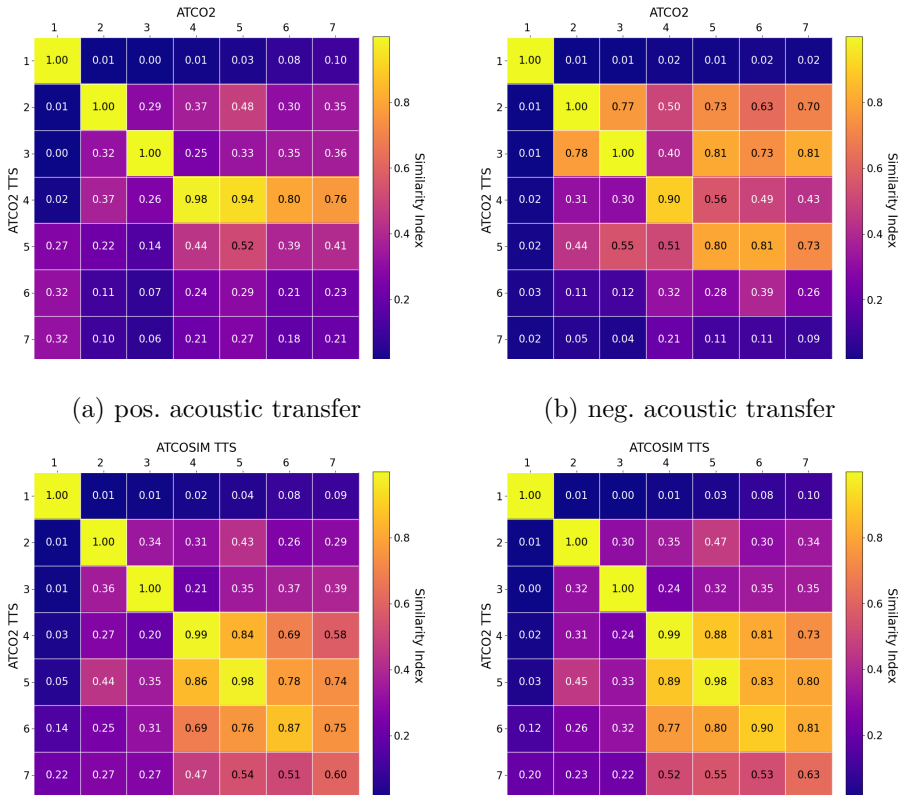
Since the influences of lexical differences and noise variability have been laid out, the question is, if there is an overall clear dependency of the WER on the ratio between the lexical differences and the noise differences. To evaluate this, we plot the WER in dependence of the ratio between the source-target LM perplexity and the source-target SNR-ratio. The resulting Figure 4.4 shows that the aforementioned dependency exists. This explains the different WERs reached on the datasets, depending on the selection of the training and test set. It furthermore opens the door for future research on predicting the WER for

unknown (ATC) benchmark datasets. While we have focused mostly on dataset features until now, we will look also at wav2vec 2.0 features in the next section.

4.4.3 wav2vec adaption

To better understand how wav2vec 2.0 adapts to the lexical differences and the acoustic variability between the different ATC datasets, we use CKA to compare the features of the different parts of the model. We examine four different cases. In the first two cases, we look at the feature similarity between two models, when one of them encounters a dataset with new acoustic properties during testing. For the positive acoustic transfer, we compare the CKA scores of wav2vec 2.0 fine-tuned on ATCO2 and ATCO2 TTS data and tested on ATCO2 TTS data. This is labeled as positive transfer case, since wav2vec 2.0 trained on ATCO2 reaches a WER of 21.5% on the unseen ATCO2 TTS data, which is a significant decrease from the 33.4 % WER on ATCO2 test data. For the negative acoustic transfer, the wav2vec 2.0 model fine-tuned on ATCO2 TTS data encounters a new dataset. Wav2vec 2.0 trained on ATCO2 TTS reaches a WER of 5.6 % on ATCO2 TTS test data but the score increases about a factor of 17 to a WER of 96.7% on the unseen ATCO2 dataset.

Figure 4.5 shows the CKA similarity scores of the wav2vec 2.0 feature encoder in the positive (a), respectively, negative acoustic transfer case (b).Interestingly, the initial and intermediate layers show even a higher similarity for the negative acoustic transfer case. But the similarity score on the final layer of the feature encoder reaches 0.21 in the positive scenario, while for the negative scenario, the similarity score is considerably lower with just 0.09. This difference also propagates through the dense transformer encoder layers as Figure 4.6 (a) and (b) show. Even in the first layer of the transformer encoder, the scores differ already significantly with 0.95 and 0.69. Towards the final layer, the difference further increases. Additionally, the CKA plot of the negative acoustic transfer does not show the typical clusters of similar representations, which can be found along the



(a) pos. acoustic transfer

(b) neg. acoustic transfer

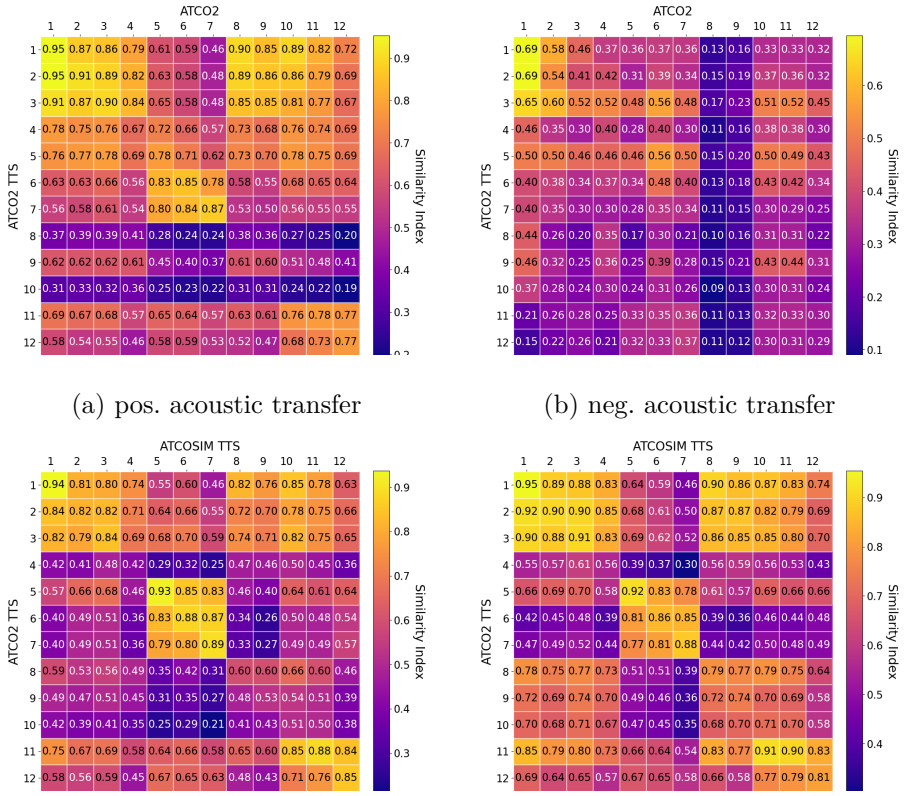
(c) pos. lexical transfer

(d) neg. lexical transfer

Figure 4.5: CKA analysis on the adaptation of the wav2vec feature encoder to acoustic and lexical changes. The CKA scores in (a) are produced on ATCO2 TTS data and the scores on (b) on ATCO2 data. The CKA scores in (c) are produced on ATCOSIM TTS data and the scores in (d) ATCO2 TTS data. All scores are given on the output layers of each convolutional layer of the feature encoder.

diagonal after fine-tuning, as observed by Phang et al. (Phang et al., 2021). For the positive acoustic transfer, there are three non-symmetric clusters visible. This higher similarity shows, that if wav2vec2.0 is trained on noisier data, it is still able to produce good output features on the cleaner dataset.

For the last two cases, we look at the similarity scores for the case, that one model encounters a dataset with different lexical properties during testing. To exclude acoustic influences, we purely use TTS data. For both, negative and positive lexical transfer, we plot the CKA similarity scores for wav2vec 2.0 fine-



(a) pos. acoustic transfer

(b) neg. acoustic transfer

(c) pos. lexical transfer

(d) neg. lexical transfer

Figure 4.6: CKA analysis: Adaptation of the wav2vec transformer encoder layers to acoustic (a) and (b) and language changes (c) and (d). The test datasets are equal to Figure 4.5.

tuned on ATCOSIM TTS and ATCO2 TTS. If wav2vec 2.0 gets fine-tuned on ATCO2 TTS, the WER on ATCO2 TTS is 5.6%, while the WER on ATCOSIM TTS is 17.4%, which is an increase of a factor of 3, but still an above average WER for an ATC dataset as Table 4.2 and Figure 4.1 show. We therefore use this scenario as positive lexical transfer. In contrast, if wav2vec 2.0 gets fine-tuned on ATCOSIM TTS, the WER on ATCOSIM TTS is 2.1%, while the WER on ATCO2 TTS is 47.0%, which is more than 20 times higher. This case is therefore labeled as negative lexical transfer. The comparison of the similarity scores of the feature encoder, Figure 4.5(c) and (d), shows that there is no significant difference between the positive and negative case. In other words, the feature encoder is agnostic to lexical differences. For the transformer encoder layers, there are evident visual

differences between the positive and negative lexical transfer. The fact, that the differences are not as big as for the acoustic transfer needs further investigation. The comparison between lexical and acoustic transfer however shows, that without the presence of noise, a cluster of similar representations in the intermediate layers of the transformer encoder is forming, which is more prominent for the positive transfer case. Since this cluster is also partially forming in the positive, but not in the negative acoustic transfer, it could be a possible candidate to indicate a good lexical and acoustic transferability.

4.5 Conclusion

Pretrained transformer-based speech recognition models, such as wav2vec 2.0, have shown remarkable performance in low-resource domains. However, for the air-traffic control domain, a highly variable transferability across different datasets has been observed. In this paper, we have presented an empirical study to identify the causes of this phenomenon. We demonstrated that each ATC dataset has specific noise characteristics. Nevertheless, adding Gaussian noise to clean air-traffic control data can be used to get a lower WER bound for different noise levels. This is an effective way to estimate the robustness of the ATC-ASR model. We have furthermore shown that there are significant lexical differences between the datasets and that the transferability correlates with cross-dataset language model perplexities as well as with the OOV rates. Dominant OOV entities are airspace-dependent cities, greetings and airlines. A target-dataset specific language model on top of wav2vec 2.0 was identified as an effective method to significantly reduce lexical mismatch and therefore the WER, even for very noise target data. With various source and target-dataset pairings, we have provided evidence for the dependency of the WER on the ratio between the source-target LM perplexity and the source-target SNR-ratio. A final wav2vec 2.0 feature analysis demonstrated, that the feature encoder is agnostic to lexical changes while adapting to different noise scenarios. Finally, we identified a same similarity cluster between the intermediate-layer-

transformer-encoder features of the target and source-data-fine-tuned wav2vec 2.0 models as indicator for a good transferability of the source-model to the target data. The insights of this work not only allow the development of better ATC-ASR models, but also better ASR models for other domains, where poor cross-dataset transferability is observed.

5

Joint vs Sequential Speaker-Role Detection and Automatic Speech Recognition for Air-traffic Control

Contents

Utilizing air-traffic control (ATC) data for downstream natural-language processing tasks requires preprocessing steps. Key steps are the transcription

of the data via automatic speech recognition (ASR) and speaker diarization, respectively speaker role detection (SRD) to divide the transcripts into pilot and air-traffic controller (ATCO) transcripts. While traditional approaches take on these tasks separately, we propose a transformer-based joint ASR-SRD system that solves both tasks jointly while relying on a standard ASR architecture. In this chapter, we compare this joint system against two cascaded approaches for ASR and SRD on multiple ATC datasets. Our study shows in which cases our joint system can outperform the two traditional approaches and in which cases the other architectures are preferable. We additionally evaluate how acoustic and lexical differences influence all architectures and show how to overcome them for our joint architecture.

The content of this chapter is based on:

> **Blatt, Alexander**, Aravind Krishnan, and Dietrich Klakow (2024). *Joint vs Sequential Speaker-Role Detection and Automatic Speech Recognition for Air-traffic Control*. URL: https://www.isca-archive.org/interspeech_2024/blatt24_interspeech.pdf.

5.1 Introduction

A standard speech processing pipeline starts with a speaker diarization (SD) module, which removes the unvoiced parts of the audio and leaves speaker labeled voiced chunks. These chunks are fed into an automatic speech recognition (ASR) system for transcription. The transcribed audio can then be further processed, for example with a natural language processing (NLP) module for information extraction. Recent architectures that combine SD and ASR show however, that they can outperform this traditional pipeline (Sarkar et al., 2018; Shafey et al.,

2019; Xia et al., 2022; Z. Huang et al., 2022; Cornell et al., 2023) by jointly utilizing acoustic and linguistic information during diarization.

The acoustic and linguistic information of air-traffic control (ATC) datasets however differs significantly from standard ASR and diarization datasets (Blatt, Abdullah, et al., 2023). ATC recordings typically have a low signal-to-noise ratio (SNR) (Blatt, Abdullah, et al., 2023) and a strict phraseology[1], which ensures an effective communication between air-traffic controllers (ATCOs) and pilots. Pilot and ATCO utterances differ in the noise level as well as in the sentence structure. This can be utilized by a SD system to differentiate between the two speaker roles `ATCO` or `PILOT`, which effectively leverages it to a speaker role detection (SRD) system.

In this work, we study how an SRD system can effectively utilize the acoustic and linguistc differences between pilot and ATCO speech by analyzing the performance, respectively robustness of different ASR&SRD architectures on multiple ATC datasets. We investigate a correlation with acoustic and linguistic properties as well as a correlation between the ASR and SRD performance. We compare three different architectures for ATC-ASR&SRD. The first method, `SRD-ASR`, consists of an acoustic-based speaker-role detection step followed by the ASR step. The second method, `ASR-SRD` first transcribes the audio before doing text-based SRD. Our proposed `Joint` method performs SRD and ASR simultaneously.

5.2 Related work

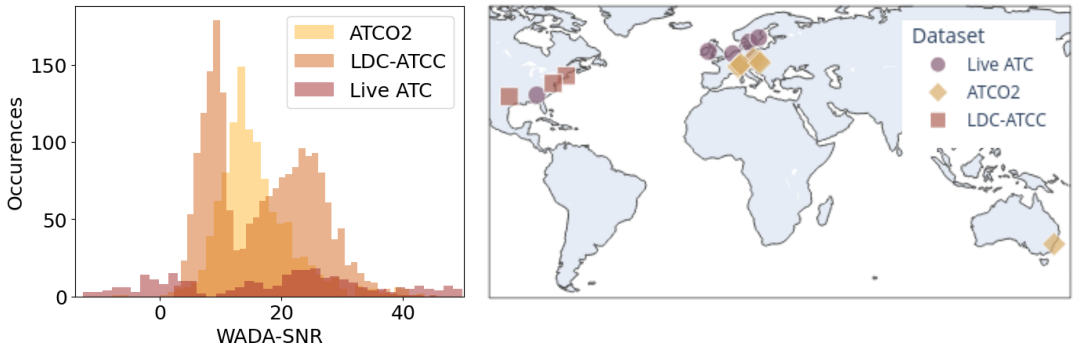
Park et al. give good general overview of speaker diarization methods (Park et al., 2022). Our `Joint` system is inspired by Shafey et al. which have first introduced a joint ASR&SD system based on a recurrent neural network transducer (Shafey et al., 2019). In contrast to Shafey et al., our system performs SRD and does not require transducers, but relies on standard transformer-based ASR models

1 ATC examples: https://wiki.flightgear.org/ATC_phraseology

(Baevski et al., 2020; Babu et al., 2022) and can be trained with traditional CTC loss (Graves, Fernández, et al., 2006b). Recent joint ASR&SD systems require even more complex architectures than the approach of Shafey et al. (Xia et al., 2022; Z. Huang et al., 2022; Cornell et al., 2023). Our text-based SRD system is based on BERTraffic (Zuluaga-Gomez, S. S. Sarfjoo, et al., 2023), which shows a 7.7 % improvement over a classical variational Bayesian hidden Markov model (VBx) (Landini et al., 2022) based approach and is to the best of our knowledge the most recent SRD model for ATC. The fact that acoustic and linguistic differences between ATC datasets negatively correlate with the performance of pretrained transformer-based ASR models has been shown by Blatt et al. (Blatt, Abdullah, et al., 2023). We investigate if there is a similar correlation for SRD.

5.3 Datasets

We use the ATCO2 (Zuluaga-Gomez, Veselý, Szöke, et al., 2022), LiveATC (Zuluaga-Gomez, Veselý, Blatt, et al., 2020a) and the LDC-ATCC corpus (Godfrey, 1994) for our experiments, since they all contain speaker labels that allow assigning each speaker either to the `ATCO` or `PILOT` class. All three corpora contain ATC communication recordings. The LDC-ATCC corpus contains solely recordings from American airports, namely Dallas Fort Worth International Airport (KDFW), Logan International Airport (KBOS) and Ronald Reagan Washington National Airport (KDCA), while the ATCO2 and LiveATC datasets contain mainly samples from European airports. They both contain samples from Václav Havel Airport Prague (LKPR) and Zurich Airport (LSZH). The ATCO2 dataset contains additionally samples from Sion Airport (LSGS), Bratislava Airport (LZIB), Bern Airport (LSZB) and Sydney Airport (YSSY) as only non-European airport. The Live ATC dataset contains additionally samples from Stockholm Västerås Airport (ESOW), Göteborg Landvetter Airport (ESGG), Dublin Airport (EIDW), Amsterdam Airport Schiphol (EHAM) and Hartsfield–Jackson Atlanta International Airport (KATL) as only American airport. All airport locations are marked in



(a) Noise Distribution (b) Geographical Distribution

Figure 5.1: Dataset dependent distributions

Figure 5.1(b). The ATCO2 corpus and the LiveATC corpus were recorded during the ATCO2 project[2]. While the ATCO2 data was recorded with VHF-receivers[3], the LiveATC corpus, consisting of the two subcorpora LiveATC1 and LiveATC2 (Zuluaga-Gomez, Veselý, Blatt, et al., 2020a), was recorded from the LiveATC webpage[4] which broadcasts ATC conversations. The ATCO2 and LiveATC dataset audio samples are recorded with a sampling frequency of 16 kHz and 16-bit, while the LDC-ATCC data is recorded with 16 kHz and 16-bit.

2 ATCO2 project: https://www.atco2.org/
3 Receiver guide: https://ui.atc.opensky-network.org/intro
4 LiveATC webpage: https://www.liveatc.net/

Table 5.1: Number of samples for the train|test|val split and the mean WADA-SNR (C. Kim et al., 2008), mean number of speaker turns and the mean (chunked) audio duration for each dataset.

Dataset	Train size	Val size	Test size	SNR (dB)	Turns	Duration (s)
ATCO2	856	107	108	15.8	2.28	9.4
LDC-ATCC	1000	500	500	16.8	3.26	13.1
LiveATC	413	42	41	18.9	2.15	12.9

Several preprocessing steps are necessary to prepare the datasets for the SRD task. To reduce the training time to a few hours per run, the original audio is chunked to samples with a target duration of 2-19 seconds. The mean chunk duration can be found in Table 5.1. This results in 2-3 speaker turns on average, as Table 5.1 shows. Samples that just contain one speaker, respectively one speaker role, are sorted out. Using the timestamps for the speaker IDs, each speaker turn is labeled with one of the two speaker roles, `ATCO` or `PILOT`. This results in transcripts, where word sequences belonging to one speaker role are tagged with either `ATCOTAG` or `PILOTTAG` as shown in Figure 5.2. For fine-tuning the ASR models, the tags are removed from the transcripts.

5.4 ASR&SRD architectures

For the ASR task of all ASR&SRD architectures, we fine-tune the Hugging Face (HF) models wav2vec 2.0[5] (w2v2) (Baevski et al., 2020) and xlsr[6] (Babu et al., 2022) on the train split of each ATC dataset. Each ASR&SRD architecture is visualized in Figure 5.2 and explained in the following sections.

5.4.1 SRD-ASR

For the SRD task of the `SRD-ASR` model we use the SD of Pyannote.audio 3.0[7] (**Bredin23**; **Plaquet23**) which combines speaker segmentation with speaker embedding-based clustering for SD. The SD tool is used out-of-the-box without further fine-tuning. Only the *max_speakers* argument is set to 2, restricting diarization to two speakers. To leverage this SD to a SRD system, the extracted speakers are matched to the speaker roles by extracting the speaker embeddings

5 HF model: facebook/wav2vec2-base-960h
6 HF model: jonatasgrosman/wav2vec2-large-xlsr-53-english
7 HF model: pyannote/speaker-diarization-3.0

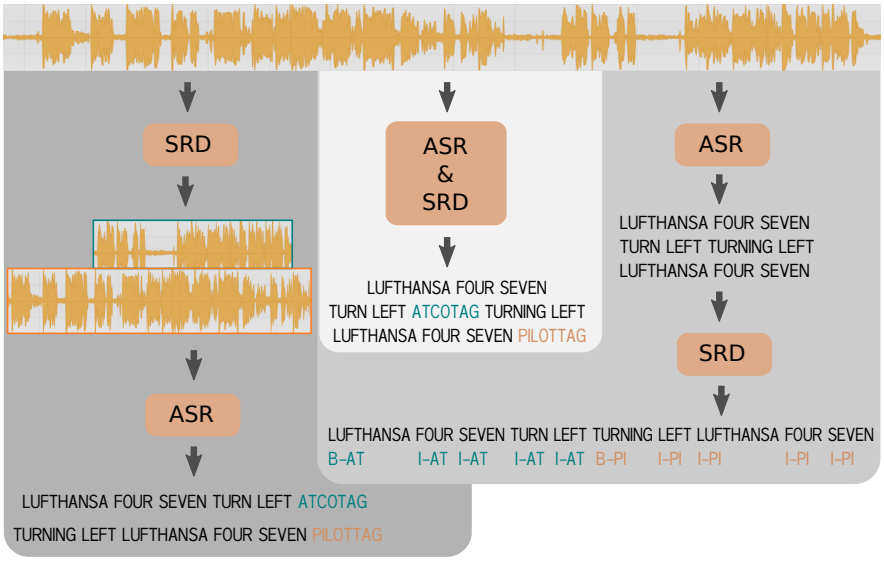


Figure 5.2: ASR&SRD architectures; left: acoustic SRD followed by ASR (`SRD-ASR`); center: Joint ASR&SRD (`Joint`); right: ASR followed by linguistic-based SRD (`ASR-SRD`)

of the identified speaker with the Pyannote speaker embedding extraction model[8]. The classification into `PILOT` and `ATCO` is done via measuring the cosine similarity between the speaker embeddings and the cluster centers of the two speaker roles for the current training data set. The cluster centers are extracted with a nearest centroid classifier[9] for each training data set by randomly selecting 50 samples for `PILOT` and `ATCO`. The speaker role-tagged utterance chunks that are produced by the SRD system are then fed into the ASR model to generate the tagged transcripts.

5.4.2 ASR-SRD

In this approach, we train a text-based diarizer using token-level speaker labels, similar to (Zuluaga-Gomez, S. S. Sarfjoo, et al., 2023). Each word in an utterance is assigned an `ATCO` or `PILOT` tag and a binary classifier is trained to predict the

8 HF model: pyannote/embedding
9 scikitlearn: Nearest centroid classifier

tag of each token. We encode `ATCO` and `PILOT` tags consistently across utterances to let the model learn speaker roles in addition to speaker turns. For training, ground-truth transcripts from the train set are used. Testing is done on the ASR transcripts generated from the test audio.

5.4.3 Joint

In the `Joint` approach, the ASR models are directly fine-tuned on the speaker role-tagged transcripts instead of transcripts without tags. Since fine-tuning is done without modifying the CTC-loss function, this approach can be applied to any transformer-based ASR model with CTC loss.

5.5 Experimental setup

All experiments are performed on an NVIDIA V100 GPU. The ASR and the `Joint` model are trained for 2000 steps, 1000 warm-up steps, a learning rate of 4e-4 and a batch size of 4 and 8 gradient accumulation steps. We choose steps instead of epochs to ensure the same number of training steps despite different sized training sets. ASR fine-tuning takes roughly 4-5 hours with these parameters. The text-based diarizer is a BERT[10] (Devlin, M. Chang, et al., 2019) model with a binary classification head on top. It is trained with a learning rate of 2e-5, 25 warm-up steps and a batch size of 16. Training is terminated with an early stopping mechanism, with a patience of 5. The models with the the lowest WER (for ASR), respectively word diarization error rate (WDER) (for SRD) on the validation data are used for testing. The WDER is implemented based on Shafey at al. (Shafey et al., 2019). We additionally measure the position error rate (PER) of the speaker role tokens. This PER allows us to measure if a speaker role token

10 HF model: https://huggingface.co/google-bert/bert-base-uncased

in placed in the correct position in the sentence independently of the speaker role. We define the PER as follows:

$$\text{PER} = 1 - \frac{t_c}{t_p} \tag{5.1}$$

where t_p is the number of ground truth speaker role tag positions and t_c is the number of correctly placed tokens at all ground truth speaker role tag positions (class independent).

High WERs can result in missing parts of the transcripts, which not only influences the WER but also the WDER and PER. To uncouple these, we align the target and predicted transcript with the Needleman-Wunsch algorithm (Needleman et al., 1970) and add placeholder tokens for non-transcribed words before calculating the WDER and PER. All experiments given in the following section are repeated thrice with different seeds and the mean and standard deviation are given over those three runs if not mentioned otherwise.

5.6 Results

5.6.1 Inter-and intra-dataset evaluation

The ASR&SRD models are tested in an inter-dataset scenario, where the train and test splits come from different datasets and an intra-dataset scenario, with the train and test splits are from the same dataset. The fact that pretrained transformer-based ASR models are susceptible to inter-dataset acoustic and linguistic variabilities (Blatt, Abdullah, et al., 2023) allows to investigate the WDER and PER scores over a wide range of WERs. The inter-dataset scores in Table 5.2 show that the `ASR-SRD` architecture outperforms the other architectures on the ATCO2 and LDC-ATCC dataset in terms of WDER when the wave2vec 2.0 model is used. On the LiveATC dataset, the `SRD-ASR` model reaches the lowest WDER despite having the highest WER. Switching from wave2vec 2.0 to xlsr results in the lowest WDER for the `Joint` model on ATCO2. The `Joint` model

Table 5.2: Inter-dataset scores: WDER,PER and WER in case the models are fine-tuned and tested on different datasets. Mean values over three runs and two training datasets are given with the standard deviation in brackets

Architecture	ASR	ATCO2			LDC-ATCC			LiveATC		
		WDER	PER	WER	WDER	PER	WER	WDER	PER	WER
SRD-ASR		38.0 (0.5)	32.8 (4.0)	71.7 (4.2)	42.9 (0.1)	37.3 (8.8)	68.5 (2.7)	**32.0** (0.6)	52.2 (4.2)	82.1 (4.1)
ASR-SRD	w2v2	**37.4** (0.4)	70.5 (1.0)	**69.4** (1.5)	**40.7** (1.0)	71.7 (1.7)	**58.7** (1.0)	39.8 (1.1)	71.4 (3.9)	**72.5** (2.2)
Joint		39.1 (4.2)	**19.8 (2.1)**	70.0 (1.1)	63.4 (3.8)	**13.8 (1.9)**	64.8 (1.0)	38.5 (10.3)	**46.3 (4.0)**	76.7 (3.0)
SRD-ASR		38.6 (0.5)	31.5 (5.1)	66.9 (5.1)	43.0 (0.3)	35.7 (9.2)	64.6 (9.1)	**31.3** (0.5)	52.0 (3.9)	76.5 (3.9)
ASR-SRD	xlsr	36.7 (0.7)	68.4 (1.7)	**60.8** (0.8)	**39.0** (0.8)	70.8 (2.0)	**53.3** (1.4)	37.8 (2.5)	69.1 (1.8)	**65.3** (1.4)
Joint		**36.6** (4.3)	**16.9** (1.7)	61.3 (1.9)	57.9 (3.1)	**10.6 (2.8)**	59.7 (1.1)	33.8 (2.3)	**41.3 (1.3)**	71.0 (2.2)

also benefits the most from the model change in the other metrics. Regarding the WER, the `ASR-SRD` model outperforms the others on all datasets, while the `Joint` model has the lowest PER score on all datasets by a margin.

This holds also true for intra-dataset scores, as shown in Table 5.3. The `Joint` model additionally has the lowest WDER on all datasets, for both the xlsr and the wave2vec 2.0 model. Although the WER scores of the `Joint` and `ASR-SRD` architecture are close in all datasets, the `ASR-SRD` model still reaches the lowest WERs in all scenarios tested. In contrast to the `SRD-ASR` architecture, the other two ASR&SRD models can more than half their WDER scores on the ATCO2

Table 5.3: Intra-dataset scores: WDER,PER and WER in case the models are fine-tuned and tested on the same dataset. Mean values over three runs are given with the standard deviation in brackets

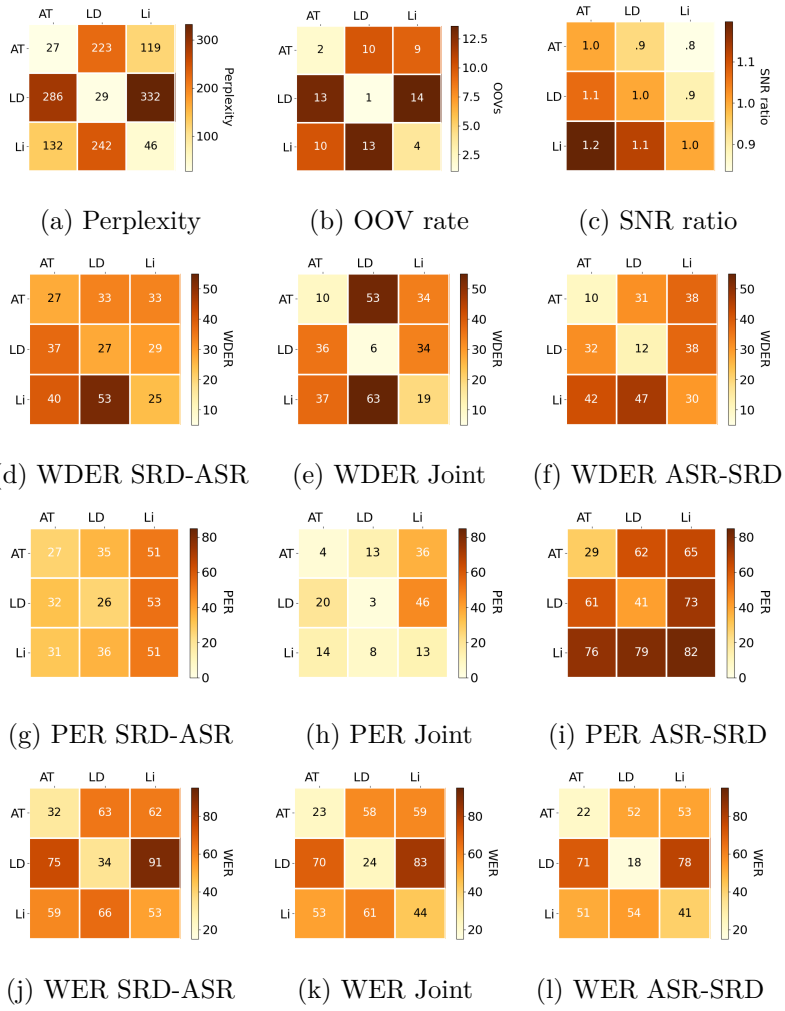
Architecture	ASR	ATCO2			LDC-ATCC			LiveATC		
		WDER	PER	WER	WDER	PER	WER	WDER	PER	WER
SRD-ASR		27.4 (0.4)	25.5 (7.9)	34.8 (3.1)	27.4 (0.1)	27.2 (5.0)	36.2 (3.1)	23.7 (0.4)	51.0 (3.6)	55.8 (4.3)
ASR-SRD	w2v2	11.4 (0.8)	33.5 (3.3)	25.9 (0.4)	12.6 (0.2)	42.1 (1.0)	**20.2** (0.1)	30.7 (0.5)	80.8 (1.6)	**43.3** (0.4)
Joint		**6.5** (0.4)	**4.8** (0.8)	**24.1** (0.3)	**8.0** (1.1)	**9.3** (0.5)	27.5 (1.1)	**19.2** (7.3)	**9.3** (0.5)	45.5 (1.8)
SRD-ASR		27.4 (0.5)	26.6 (9.0)	32.0 (3.3)	27.5 (0.3)	25.9 (4.7)	34.4 (1.6)	24.7 (0.4)	50.7 (2.1)	53.1 (0.8)
ASR-SRD	xlsr	10.4 (0.2)	28.6 (3.4)	**22.1** (0.6)	12.3 (0.3)	41.0 (0.1)	**17.6** (0.4)	30.2 (0.2)	81.8 (0.6)	**41.0** (0.2)
Joint		**9.9** (3.0)	**4.0** (1.0)	23.1 (1.5)	**6.2** (0.3)	**2.6** (0.3)	23.9 (1.1)	**19.1** (0.5)	**12.7** (2.9)	43.5 (0.9)

and LDC-ATCC dataset compared to the inter-dataset scenario. This indicates that they can utilize the fact that the lexical features do not change significantly between the training and testing scenario. This is further analyzed in the next chapter.

5.6.2 Relation and causation analysis for ASR&SRD

To decouple/correlate the ASR and SRD performance, we analyze the confusion matrices for WDER, PER and WER in Figure 5.3. Additional matrices for the out-of-vocabulary (OOV) rates and the perplexities allow us to draw a connection to linguistic differences between the datasets. The perplexities are calculated by building a 4-gram language model (LM) on the training data and calculating the perplexity with this LM on the test data. The acoustic influences can be investigated by analyzing the SNR train/test ratio confusion matrix. The SNR values are estimated with the WADA-SNR algorithm (C. Kim et al., 2008).

All three architectures have similar WER confusions matrices, the fact that `SRD-ASR` transcribes already speaker role chunked audio files just shows in the absolute values. The PER matrices differ however significantly. The `SRD-ASR` model seems to mostly underperform when tested on the LiveATC dataset while the `ASR-SRD` model produces high PERs when trained on the LiveATC dataset. The `Joint` model shows a balanced performance except for the case when trained on LDC-ATCC and tested on LiveATC. This corresponds with the perplexity and OOV rate matrices, which also show a high value for this pairing.



(a) Perplexity (b) OOV rate (c) SNR ratio

(d) WDER SRD-ASR (e) WDER Joint (f) WDER ASR-SRD

(g) PER SRD-ASR (h) PER Joint (i) PER ASR-SRD

(j) WER SRD-ASR (k) WER Joint (l) WER ASR-SRD

Figure 5.3: Confusion matrices for different metrics (a)-(l) and different ASR&SRD methods (d)-(l) run with the xlsr model. The columns correspond to the test datasets and the rows to the training dataset. The SNR train/test ratio is calculated based on the values of Table 5.1. The datasets are abbreviated as follows: AT: ATCO2, LD: LDC-ATCC, Li: LiveATC.

The WDER matrices show that the `SRD-ASR` architecture has the most balanced performance while only producing high WDERs on the liveATC - LDC-ATCC pairing, which is also the case for the other architectures. This could be due to the fact that this pairing shows also a high perplexity, OOV rate and SNR ratio. The confusion matrix on the `Joint` model highlights the performance gap between the inter and intra-dataset scenario. The WDER, WER and PER matrices of the `ASR-SRD` value show a high similarity, indicating a correlation between the

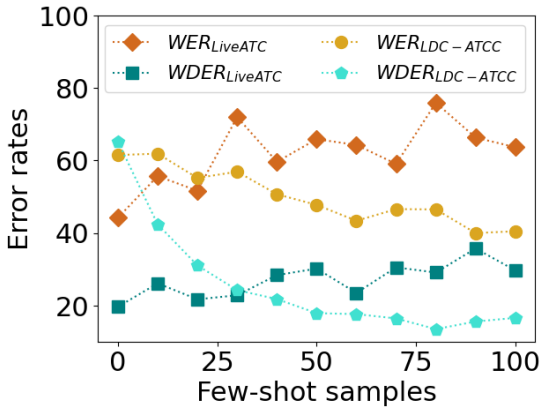


Figure 5.4: Few-shot learning on LDC-ATCC of a `Joint`-xlsr model finetuned previously on Live ATC data. All experiments are just conducted once.

three measures. Overall, the perplexity and OOV rates seem to have a higher influence on the ASR&SRD metrics than the SNR ratio. But it should be noted, that the WADA-SNR values of the datasets are quite similar as Table 5.1 shows. However, the distribution of the SNR values are quite different as Figure 5.1 (a) indicates. An additional noise analysis is therefore necessary to draw noise-related conclusions.

5.6.3 Few-shot learning

The difference between the inter- vs intra-dataset WDER scores for the `Joint` architecture is quite large as shown above. To ameliorate this with domain familiarization, we resort to few-shot training. As an example case, we use the LDC-ATCC data to further train a `Joint`-xlsr model finetuned on Live ATC data. Figure 5.4 shows that the WDER on LDC-ATCC drops to 42% by just using 10 samples from the LDC-ATCC data for fine-tuning. This is already the level that the other architectures reach on this dataset. By using 50 samples, the WDER is already below 20%. There is however a noticeable increase in the WDER/WER on the Live ATC dataset. At 25 samples, the model shows a balanced inter-and intra-dataset

performance. This shows that adaptation to the cross-dataset scenario is possible by using few-shot training.

5.7 Conclusion

Recently proposed joint diarization and ASR models outperform traditional sequential approaches. The air-traffic control (ATC) domain differs however acoustically and linguistically from standard diarization and ASR datasets. In ATC, identifying the speaker role, pilot or air-traffic controller, is often more important than identifying the speaker. We have therefore proposed a joint speaker-role detection (SRD) and ASR system for ATC (`Joint`). This system purely relies on transformer-based ASR models. We have compared this architecture against two traditional cascaded approaches, which either first perform ASR, then text based SRD (`ASR-SRD`), or first acoustic-based SRD and then ASR (`SRD-ASR`). Our system clearly outperforms the other systems in the intra-dataset scenario in terms of the word diarization error rate (WDER). The position error rate (PER) scores are lower in all scenarios. We can show that the WDER scores of the (`Joint`) and (`ASR-SRD`) systems scale with a better ASR performance, while the (`Joint`) models seems to benefit more from a potent ASR model. Few-shot training results indicate that the inter-dataset scores of the `Joint` model can be significantly improved with just 25 samples. The `ASR-SRD` architecture shows a more balanced performance between the intra- and inter-dataset scenario, while the `SRD-ASR` approach only seems to be superior if there is a high WER scenario. These insights allow to pick the correct architecture for an individual ASR&SRD task.

6

Call-sign Recognition and Understanding for Noisy Air-traffic Transcripts Using Surveillance Information

Contents

Air-traffic control (ATC) relies on communication via speech between pilot and air-traffic controller (ATCO). The call sign, as a unique identifier for each flight, is used to address a specific pilot by the ATCO. Extracting the call-sign from the communication is a challenge because of the noisy ATC voice channel

and the additional noise introduced by the receiver. A low signal-to-noise ratio (SNR) in the speech leads to high word error rate (WER) transcripts. In this chapter, we propose a new call-sign recognition and understanding (CRU) system that addresses this issue. The recognizer is trained to identify call-signs in noisy ATC transcripts and convert them into the standard International Civil Aviation Organization (ICAO) format. By incorporating surveillance information, we can multiply the call-sign accuracy (CSA) up to a factor of four. The introduced data augmentation adds additional performance on high WER transcripts and allows the adaptation of the model to unseen airspaces.

The content of this chapter is based on:

Blatt, Alexander, Martin Kocour, Karel Veselý, Igor Szöke, and Dietrich Klakow (2022). "Call-Sign Recognition and Understanding for Noisy Air-Traffic Transcripts Using Surveillance Information." In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8357–8361. DOI: 10.1109/ICASSP43922.2022.9746301.

6.1 Introduction

The classical communication between air-traffic controllers (ATCOs) and pilots is voice-based (Eskilsson et al., 2020). This form of communication has the drawback, that one ACTO talks to multiple pilots over a single frequency. The rising traffic in the last years increased the number of pilots tuned in the same frequency. This increases the chance, that two pilots speak simultaneously. To avoid responses from multiple pilots, the ATCO addresses the target airplane by its call-sign. A call-sign is a unique identifier that is assigned to each airplane (e.g. `DLH83K`). New systems like controller–pilot data link communications (CPDLC), which use text-based communication, reduce the load on the voice communication channels (Eskilsson

et al., 2020). Projects like AcListant and MALORCA[1] aim to support the ATCO by speech recognition systems (Kleinert, Helmke, Siol, Ehr, Cerna, et al., 2018; Srinivasamurthy et al., 2018). The problem with developing such systems, is the lack of training data in the ATC domain. Although there exist some datasets (Zuluaga-Gomez, Motlicek, et al., 2020), a database is missing which covers a multitude of locations and contains speech, transcripts, and meta-information like call-signs and commands. This work is part of the ATCO2 project[2], which aims, among others, to build up such a database.

In this work we are investigating the benefit of including context information for call-sign recognition and understanding. The context information in the form of a list of surveillance call-signs is used as an additional input for our models. The models recognize the call-sign in an ATC transcript and convert it to the standard ICAO format. For the training of our models, we introduce a data augmentation method, that is adjustable to the target airspace. We can show, that the models trained on the augmented data predict the target call-signs with high accuracy. We also find that the models which are incorporating surveillance information are superior and show a high resistance to ASR noise and surveillance data variations.

6.2 Related Work

Various works have already investigated context incorporation in ASR (Shore et al., 2012b; Schmidt et al., 2014; Oualil et al., 2015), which marks the prior step in the ATC speech processing pipeline. Two other works of the ATCO2 project (Kocour, Veselý, Blatt, et al., 2021; Nigmatulina et al., 2021) show that the combination of HCLG and lattice boosting using Kaldi (Povey et al., 2011), reduces the ATC-ASR errors, especially for the call-signs. We build on top of these works by extracting the (erroneous) call-signs from the ASR transcripts and map them to the standardized ICAO format.

1 MALORCA Homepage: https://www.malorca-project.de/
2 ATCO2 Homepage: https://www.atco2.org/

Table 6.1: Overview of the datasets. The last column marks the WER of the different versions of the same dataset.

Datasets	Samples	WER Variants
LiveATC	500	0|28.4 (h)|28.9 (l)|33.1 (b)
Malorca	1130	0|6.42 (h)|7.27 (l)|8.47 (b)
Airbus	60000	0|7.00|30.0

In named-entity recognition (NER) the call-sign sequence is identified in the input (Recognition), therefore it is related to our method, which additionally converts the call-sign to the target ICAO format (Understanding). NER for call-signs as single entity of interest is also part of the Airbus challenge (Pellegrini et al., 2019). One of the top three contestants uses a Bi-LSTM-CRF architecture (V. Gupta et al., 2019) for the call-sign recognition, reaching an F1 score of 80.17 on the leader board. Newer pretrained transformer-based models like BERT typically outperform recurrent architectures like LSTMs in natural language processing (NLP) and natural language understanding (NLU) tasks (Devlin, M. Chang, et al., 2019).

6.3 Data

Table 6.1 contains the datasets, that are used for training and testing. The Malorca dataset (Kleinert, Helmke, Siol, Ehr, Cerna, et al., 2018; Srinivasamurthy et al., 2018) consists of ATCO speech transcripts from the Vienna airport together with surveillance call-signs for each transcript. The LiveATC dataset contains transcripts of ATC speech from Zurich Airport (LSZH) and Dublin Airport (EIDW) with some samples from Hartsfield–Jackson Atlanta International Airport (KATL). The speech data is collected during the ATCO2 project from LiveATC[3],

3 LiveATC Homepage: https://www.liveatc.net/

which provides live ATC radio feeds. The Malorca and LiveATC transcripts are generated by three different ASR methods (baseline (b), lattice-boosting (l) and HCLG-lattice boosting (h)) (Kocour, Veselý, Blatt, et al., 2021) and by human transcription for the ground-truth data (WER 0). All transcripts are manually annotated with the correct ICAO call-sign. The generation of the augmented Airbus datatset out of the Airbus development dataset (Delpech et al., 2019) is described in Section 6.4.

A sample of the datasets consists out of the transcript (`lufthansa eight three kilo descend three thousand feet`), the corresponding target ICAO call-sign (`DLH83K`) and the surveillance call-signs (`AIF44T`, `DLH83K`, `MAN47N`, ...). The surveillance data is drawn from the OpenSky Network[4] (OSN) database (Schäfer et al., 2014). We isolate the call-signs from the surveillance–broadcast (ADS-B) data fetched for each transcript and use them as context information. On average, a sample contains 26 (Malorca), respectively 30 (LiveATC) surveillance call-signs.

The call-signs start generally with an airline identifier[5] (`lufthansa` ↔ `DLH`) followed by an alphanumeric call-sign number (`eight three kilo` ↔ `83K`). The call-sign number in the transcript is converted to its ICAO equivalent by using the NATO phonetic alphabet[6].

6.4 Data Augmentation

Each airspace has distinct characteristics like the occurrence of regional airlines. The noise levels of the voice channel can also vary, resulting in different WERs of the transcripts. Ideally, a CRU system could be fine-tuned to each new airspace by training it on a database for this region. In reality, there exist only a handful

4 OSN Homepage: https://opensky-network.org/

5 A list of identifiers can be found here: https://en.wikipedia.org/wiki/List_of_airline_codes

6 NATO phonetic alphabet: https://en.wikipedia.org/wiki/NATO_phonetic_alphabet

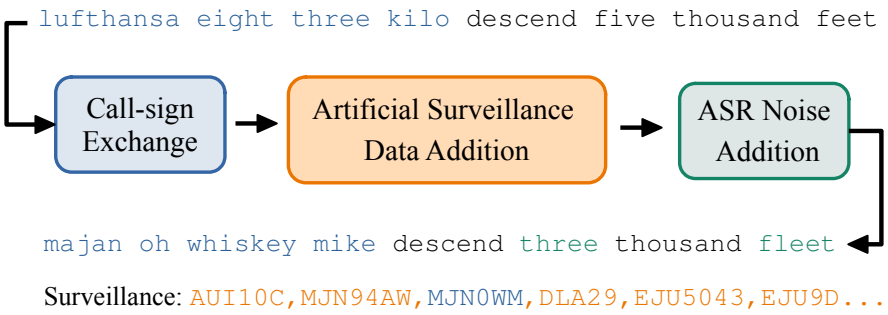


Figure 6.1: Scheme of the data augmentation pipeline.

of ATC databases (Zuluaga-Gomez, Motlicek, et al., 2020). But not all of them contain labeled call-signs. In most cases, a timestamp as well as the location of the recordings are also missing, which makes it impossible to retrieve the corresponding surveillance information from the OSN database.

To overcome this issue, we propose a data augmentation pipeline, which is shown in Figure 6.1. The basis for the pipeline is the Airbus training dataset (Delpech et al., 2019), which contains approximately 28.000 transcripts with labeled call-signs. In the first step of the augmentation, the call-sign (`lufthansa eight three kilo`) is cut out of the transcript and replaced with an artificially generated call-sign (`majan oh whiskey mike`).

The rule-based data augmentation also includes real-life variations from the standard format. This includes missing identifiers, shortened call-sign numbers and the usage of different identifier formats. Transcript equivalents of `DLH72K` are for example `lufthansa seven two kilo`, `seven two kilo`, `lufthansa`, `lufthansa seventy-two kilo` and `dlh seven two kilo`.

In the next step, surveillance call-signs are added with the same parameters (number of call-signs with the same identifier, number of total call-signs, surveillance length) as real surveillance. To match the noise level of the test datasets (Malorca and LiveATC), simulated ASR noise (noisy distribution extracted from noisy ASR output) is introduced in the last step for the two noisy datasets (WER 7.0 and WER 30.0) but not for the clean dataset (WER 0).

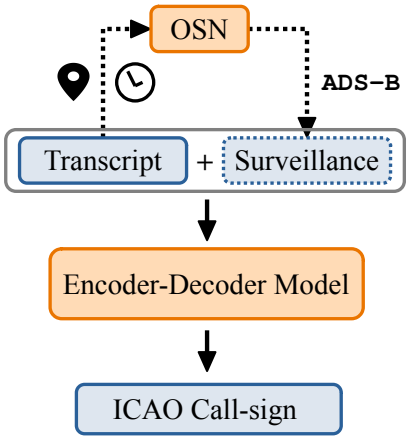


Figure 6.2: The CRU system. The dotted path marks the optional surveillance retrieval via OSN with the aid of the transcripts timestamp and VHF receiver location.

6.5 Context Integration

Context integration is necessary, since not all of the information loss through the ASR system can be recovered. If for example `five` would be missing in `Ryanair eight five three kilo`, the remaining `Ryanair eight three kilo` would be the wrong, but valid call-sign. A conventional CRU system would therefore predict `RYR83K` as target call-sign instead of `RYR853K`. Adding surveillance call-signs as additional input, as shown in Figure 6.2, allows the model to compensate the missing information in the transcript. With the timestamp of the transcript and its recording location, surveillance information can be retrieved from OSN (dotted path in Figure 6.2), like described in Section 6.3.

6.6 Experimental Setup

The basis of our CRU model is the `EncoderDecoderModel` from the Hugging Face transformers library (Wolf et al., 2020), which showed superior performance over other designs in prior experiments. The language model head of this architecture

allows also to predict call-signs without surveillance information. For both, encoder and decoder, the pretrained `bert-base-uncased` model is used, to make use of the beneficial effect of using pretrained models for sequence-to-sequence tasks (Rothe et al., 2020).

Since ATC speech transcripts differ highly from standard text, a domain adaptation is performed by pretraining BERT (`bert-base-uncased`) on ATC transcripts using masked language modeling. The CRU models are trained on the augmented Airbus datasets. For each augmented dataset listed in Table 6.1 (WER 0, WER 7 and WER 30) a split of 40k/10k/10k for train/val/test sets is used. The models are either trained with (Sur) or without (Van) surveillance information. The transcript and the surveillance call-signs are concatenated and embedded into a single vector. This single or cross-encoder design allows interactions between the transcript and context from lower layers of the model on. The overall architecture for the Sur and Van models is the same, to ensure a fair comparison. The trained models are tested on the LiveATC and Malorca test sets listed in Table 6.1. The performance of all models is measured as accuracy or call-sign accuracy (CSA).

6.7 Experimental Results

6.7.1 Surveillance Incorporation

Feeding the model surveillance call-signs not only allows recovering noisy ASR transcripts, that are lacking e.g. the airline identifier. The surveillance allows the model also to predict call-signs containing airline identifiers, that did not appear in the training data. Additionally, the surveillance call-signs decrease the target space for the model.

Table 6.2 and Table 6.3 show the comparison between CRU models incorporating surveillance (Sur) and not incorporating surveillance (Van). The models that include surveillance call-signs outperform the vanilla models on every test set. On

Table 6.2: Accuracy on the LiveATC test sets. The call-sign recognition models are trained on the augmented Airbus dataset with different WERs. Underlined accuracy scores symbolize the best vanilla recognition model, while bold scores mark the best model overall.

Taining sets	Accuracy on LiveATC test sets							
	WER 0		WER 28.4		WER 28.9		WER 33.1	
	Van	Sur	Van	Sur	Van	Sur	Van	Sur
WER 0	39.8	**89.4**	31.0	74.0	27.8	70.3	11.2	45.2
WER 7	<u>60.2</u>	88.6	<u>47.2</u>	**78.4**	<u>44.0</u>	**73.3**	<u>16.4</u>	**57.0**
WER 30	56.0	85.6	46.6	73.6	42.8	68.5	15.4	47.4

the high-noise transcripts of the LiveATC dataset (WER 33.1), the benefit of the additional information shows the best. The vanilla network is here outperformed by a factor of 3-4. As an example of the recovery capabilities for noisy call-signs, we are able to predict 57% of the ICAO call-signs from the LiveATC transcripts (WER 33.1). Although they contain only 27% correct call-signs. This means an increase of 30%.

6.7.2 Noisy Training Data

To give the models more robustness against ASR noise, they are trained on different WER-level training data. On the Malorca data, the models trained on noisy transcripts (WER 7 and WER 30) outperform the model trained on clean transcripts (WER 0) on every test set as Table 6.3 shows. Both, the surveillance and the vanilla models benefit on similar levels from the training on noisy data, while the highest performance boost is reached on the noisiest test set (WER 8.47) from 75.6% accuracy to 81.0%. On the noisier LiveATC test sets, the overall mean accuracy of the vanilla model trained on WER 7 data is around 1.5 times higher than the accuracy of the model trained on noise-free data as stated in Table 6.2.

Table 6.3: Accuracy on the Malorca test sets. The call-sign recognition models are trained on the augmented Airbus dataset with different WERs. Underlined accuracy scores symbolize the best vanilla recognition model, while bold scores mark the best model overall.

Taining sets	Accuracy on Malorca test sets							
	WER 0		WER 6.42		WER 7.27		WER 8.47	
	Van	Sur	Van	Sur	Van	Sur	Van	Sur
WER 0	49.5	85.6	50.6	82.4	47.4	79.5	44.2	75.6
WER 7	53.8	**87.5**	53.6	84.9	50.4	83.5	46.8	80.7
WER 30	<u>54.8</u>	87.3	<u>54.7</u>	**85.0**	<u>50.9</u>	**83.7**	<u>47.2</u>	**81.0**

Raising the WER of the training data further from 7 to 30 leads only to a small improvement on the high WER Malorca test sets.

The results show the benefit of training the model on (simulated) noisy transcripts if the target input of the model is the output of the ASR recognizer. But more importantly, they also show, that even if there is just clean data available for training, including the surveillance call-signs is a necessary condition to reach maximum performance.

6.7.3 Surveillance Fluctuation Robustness

We investigate the robustness of our model against the three main surveillance parameters: the number of call-signs in the surveillance, the number of airline identifier duplicates and the number of call-sign number duplicates. The evaluations are done on the LiveATC dataset by altering the surveillance information. The model trained on the WER 7 dataset is used for these tests, since it performs the best on the noisy LiveATC test sets.

A higher number of surveillance call-signs increases the search space for the model. By increasing the surveillance size from 1 to 19, the accuracy decreases

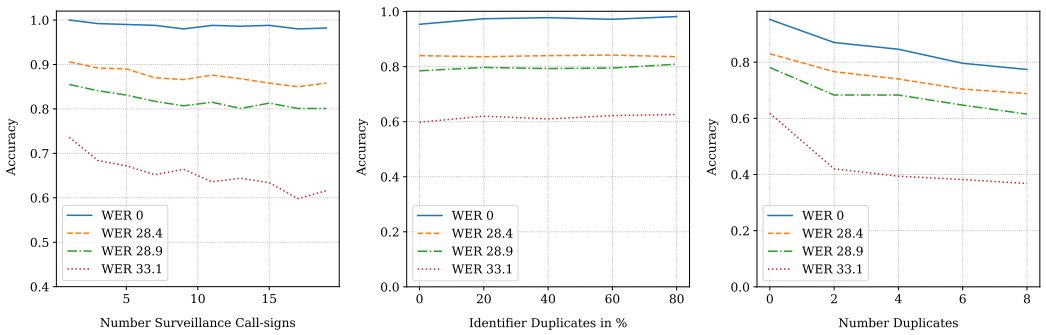


Figure 6.3: Change of accuracy depending on (left) the number of call-signs in the surveillance data; (middle) the relative number of additional call-signs in the surveillance information containing the same call-sign identifier as the target call-sign; (right) the number of additional call-signs in the surveillance information containing the same call-sign number as the the target call-sign.

by 5% on the WER 28.4 test data, while there is a decrease of 12% on the WER 33.1 test set as Figure 6.3 shows. Intuitively, this is clear since on noisy data, the model has to rely more on the additional context information.

Several airplanes of the same airline can be in the same airspace resulting in call-signs with an identical identifier (e.g. `DLH124, DLH9M, DLH69F`). For the LiveATC and Malorca test set, each identifier in the surveillance occurs 1.45 respectively 1.9 times. Figure 6.3 shows, that the recognizer is very robust against airline identifier duplicates. Even with 80% of the surveillance call-sign identifiers being identical to the target identifier call-sign, there is no drop in accuracy.

In contrast to identifier duplicates, having the same call-sign number in the surveillance information (e.g. `DLH83K, CSA83K, RYR83K`) is quite rare. In the LiveATC dataset, only in 2.7% of the cases, a call-sign number appears twice in the surveillance information. With one duplicate of the target call-sign, which is already higher than what can be expected in the real-life scenario, the accuracy drop on the WER 28.4 and WER 28.9 dataset stays below 5% as Figure 6.3 shows. For the high-noise dataset, the drop is around 10%.

6.8 Conclusion

In this work, we have introduced a method for enhancing call-sign recognition and understanding (CRU) by incorporating context information in the form of surveillance call-signs without changing the model architecture. We have shown that this improves the call-sign accuracy up to 4 times. Our data augmentation pipeline allows to generate training data for specific airspaces, even if there are no transcripts available for that region. We have shown that introducing ASR noise in the data augmentation pipeline improves the vanilla model performance up to 1.5 times.

We can show that our models are robust against the occurrence of multiple surveillance call-signs containing the same identifier. The number of included surveillance callsigns should be kept as low as possible since the call-sign accuracy decreases linearly with the number of the surveillance call-signs. For the rare case of an additional call-sign occurring with the target call-sign number, we can show that the accuracy drop stays below 5% for the low call-sign WER test sets and under 10% for the high WER call-sign test set.

In the future, we want to also look at other context incorporation methods. We additionally plan to adapt our model to other named entities appearing in ATC transcripts, such as commands and values.

7

Utilizing Multimodal Data for Edge Case Robust Call-sign Recognition and Understanding

Contents

Operational machine learning-based assistant systems must be robust in a wide range of scenarios. This holds especially true for the air-traffic control (ATC) domain. The robustness of an architecture is particularly evident in edge

cases, such as high word error rate (WER) transcripts resulting from noisy ATC recordings or partial transcripts due to clipped recordings. In this chapter, we therefore specifically focus on edge cases to get better insight on model robustification for ATC. To increase the edge-case robustness of call-sign recognition and understanding (CRU), a core tasks in ATC speech processing, we propose the multimodal call-sign-command recovery model (CCR). The CCR architecture leads to an increase in the edge case performance of up to 15%. We demonstrate this on our second proposed architecture, CallSBERT. A CRU model that has fewer parameters, can be fine-tuned noticeably faster and is more robust during fine-tuning than the state of the art for CRU. Furthermore, we demonstrate that optimizing for edge cases leads to a significantly higher accuracy across a wide operational range.

The content of this chapter is based on:

> **Blatt, Alexander** and Dietrich Klakow (2024). *Utilizing Multimodal Data for Edge Case Robust Call-sign Recognition and Understanding*. arXiv: 2412.20467 [cs.CL]. URL: https://arxiv.org/abs/2412.20467.

7.1 Introduction

Pilots rely on the guidance of air-traffic controllers (ATCO) for a safe take-off and landing. Research projects targeting ACTO-pilot communication automation like AcListant, Malorca (Srinivasamurthy et al., 2018) or ATCO^2 (Zuluaga-Gomez, Veselý, Blatt, et al., 2020b) are enabling the development of fully automated air-traffic control (ATC) speech processing pipelines and assistant systems. These systems should be robust and tested in edge case scenarios, which the European Union Aviation Safety Agency states specifically (European Union Aviation Safety Agency, 2021). This contrasts with the fact that the majority of the developed mod-

els are optimized for standard conditions on data sets like ATCOSIM (Hofbauer et al., 2008), AIRBUS (Pellegrini et al., 2019) or ATCO2 (Kocour, Veselý, Szöke, et al., 2022). These data sets are not designed for edge-case testing and often lack edge-case samples, like for example high noise recordings. This is problematic since high noise conditions with low SNR values are occurring during operation. If a machine learning (ML) system is not properly adapted to those conditions, hallucinations or drastic performance degradation can occur (Ji, Lee, et al., 2023).

We address this at the example of call-sign recognition and understanding (CRU) (Blatt, Kocour, et al., 2022). Extracting the call-sign from ATC speech, respectively transcripts, is one of the key tasks in ATC. ACTOs address their commands to a specific pilot by starting each instruction with a call-sign[1]. A misrecognized call-sign can lead to incidents or in the worst case accidents. Our first contribution to this topic is the introduction of CallSBERT, a novel, smaller and faster to train CRU model that can be used more flexible than the state of the art (SOTA) for CRU. As a second contribution, we show that training in edge cases like *high WER*, *clipping* and *missing transcripts* can significantly improve the accuracy not only in these edge cases but throughout the operational range. We propose the call-sign-command recovery model (CCR) which utilizes commands and plane coordinates to recover additional call-sign accuracy (CA) in edge cases and can even compensate for completely erroneous transcripts.

7.2 Related work

Related works focus on call-sign tagging (V. Gupta et al., 2019), call-sign transcription (Nigmatulina et al., 2021) or call-sign recognition in the International Civil Aviation Organization (ICAO) format from ATC conversation transcripts (Blatt, Kocour, et al., 2022; Ohneiser, S. Sarfjoo, et al., 2021). Multimodal approaches for automatic speech recognition (ASR) in ATC use surveillance call-signs from

1 ATC examples: https://wiki.flightgear.org/ATC_ phraseology

Automatic Dependent Surveillance–Broadcast (ADS-B) information to boost the performance (Guo et al., 2021; Kocour, Veselý, Szöke, et al., 2022). In an earlier work, we propose a call-sign recognition and understanding (CRU) model using surveillance call-signs (Blatt, Kocour, et al., 2022). The surveillance model in this work, called the *EncDec* model in the following, relies on ATC transcripts as input, which allows to evaluate the CRU task independently from the ASR task. This is the CRU reference model for our edge-case optimization. Plane locations are also useful context information, since commands are given from ATCOs to pilots usually at defined areas in the airspace. Kleinert et al. (Kleinert, Helmke, Siol, Ehr, Finke, et al., 2017) include plane locations via binary 2D airspace command distributions to improve their controller command prediction. We extend this idea, by using more informative non-binary 3D distributions in our command distribution module (CDM), which is one of the key components for our robust edge-case CRU performance. Our CRU model CallSBERT is based on SBERT (Reimers et al., 2019) and we adapt BERT (Devlin, M. Chang, et al., 2019) as command classifier in our edge-case robust CCR architecture.

7.3 Data preparation

The CRU models are trained on ATC transcripts of the MALORCA data set (Prague airport) and on transcripts of the AIRBUS data set. Both datasets contain ATC transcripts labeled with the correct call-signs, e.g. `ryanair one two four` (expanded format), respectively `RYR124` (ICAO format). The AIRBUS dataset, with artificial surveillance data added (Blatt, Kocour, et al., 2022), is only used for pretraining. This pretraining is crucial since the MALORCA dataset is relatively small. The train|val|test split consists of 0.9K|0.1K|0.1K samples for the MALORCA dataset, respectively 8.9K|1.3K|1.3K samples for the AIRBUS dataset. To generate samples for command classification, the data is multilabeled with a key-word-based labeler that recognizes six command types: *horizontal*,

vertical, *ils*, *taxi*, *clearing* and *greeting*. A transcript is tagged with *horizontal* if it contains for example the key words `turn right` or `change heading`.

For each of the MALORCA transcripts, the ADS-B information of each airplane in 100 km N-S and E-W distance of the Prague airport and 0-20 km altitude is fetched from the OpenSky data base[2] via the timestamp of the transcript. From the ADS-B state vectors, the coordinates of the planes in the 200 km · 200 km · 20 km bounding box are isolated and transformed to an xyz coordinate system with its origin located at the airport. Approximately 30 planes are within this bounding box at the same time. Therefore, a **random baseline** for call-sign identification has a chance of 1/30 to identify the correct call-sign. For the different edge cases in Section 7.5.2.1, Section 7.5.2.2 and Section 7.5.2.3, the transcripts are altered accordingly. Versions of different WERs are produced by adding ASR noise as described in (Blatt, Kocour, et al., 2022). Additionally, clipped versions of the transcripts are produced by removing n words from the beginning of the transcript. All experiments are run on a NVIDIA GeForce RTX 2060 GPU. All experiments are run thrice and the mean and standard deviation are given. For each run, the model with the lowest validation loss is chosen for testing.

7.4 Models

7.4.1 EncDec

As SOTA, we take the EncDec model from Blatt et al. (Blatt, Kocour, et al., 2022) which uses a *bert-base*[3] encoder-decoder architecture and has 66.3M parameters. One mayor drawback of the EncDec architecture is the way, in which the model is trained. The input of the model consists of a transcript concatenated with **all**

2 OpenSky: https://opensky-network.org/
3 Transformer library: https://huggingface.co

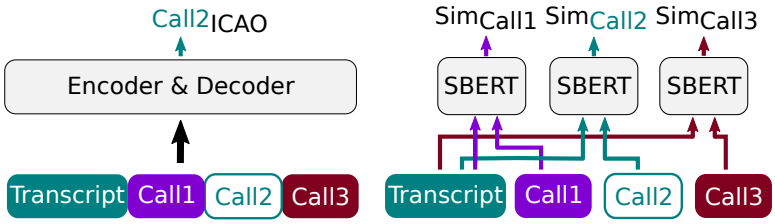


Figure 7.1: Architecture comparison of the parallel EncDec (Blatt, Kocour, et al., 2022) (left) and the sequential CallSBERT model (right).

surveillance call-signs to predict the target call-sign directly in the ICAO format as Figure 7.1a shows.

7.4.2 CallSBERT

The CallSBERT model takes the transcript and only **one** matching or non-matching surveillance call-sign for the contrastive loss training. This significantly reduces the input size. In Figure 7.1, *Call2* is an example of a matching call signal (positive sample), while *Call1* is a non-matching call-sign (negative sample). The CallSBERT architecture is based one SBERT block[4] (Reimers et al., 2019), visualized in Figure 7.1b, and has only 37.1% (24.6M parameters) of the EncDec model parameters. All this results in an increased training speed of a **factor of 4**[5] in comparison with the EncDec Model. If the models are applied to a bigger airspace, with more surveillance call-signs present, this factor will further increase. Since CallSBERT ranks the surveillance call-signs **sequentially** during inference via cosine-similarity scores (*Sim*), its maximum input size does not need to be defined beforehand, which is an advantage of this architecture. The production of similarity scores also allows this architecture to be used in a sub-model, because the similarity scores for each surveillance call-sign can be used as features. In contrast, the EncDec architecture does only predict one call-sign and gives no

4 SBERT library: `https://www.sbert.net`

5 roughly 100 s vs 400 s for 10 epochs finetuning on the 0.9K MALORCA train split

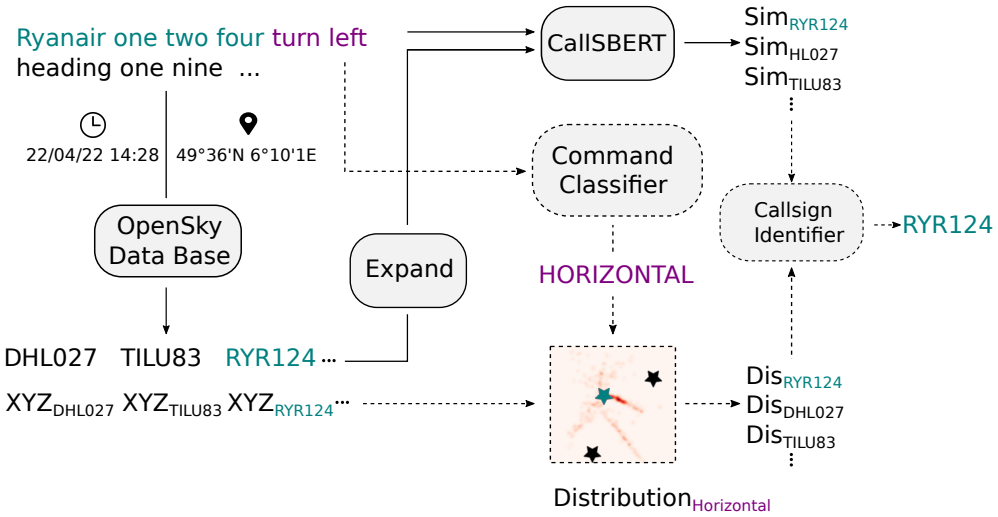
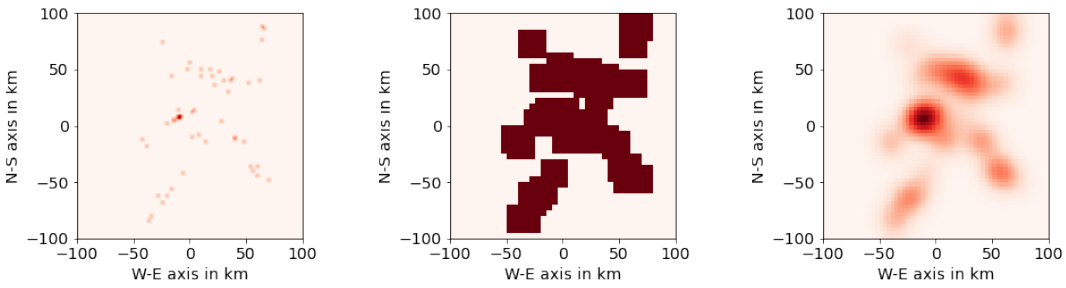


Figure 7.2: CCR architecture. The dotted lines mark the additional call-sign prediction path via command distributions.

information about the other call-signs which is the main disadvantage of this model.

7.4.3 CCR

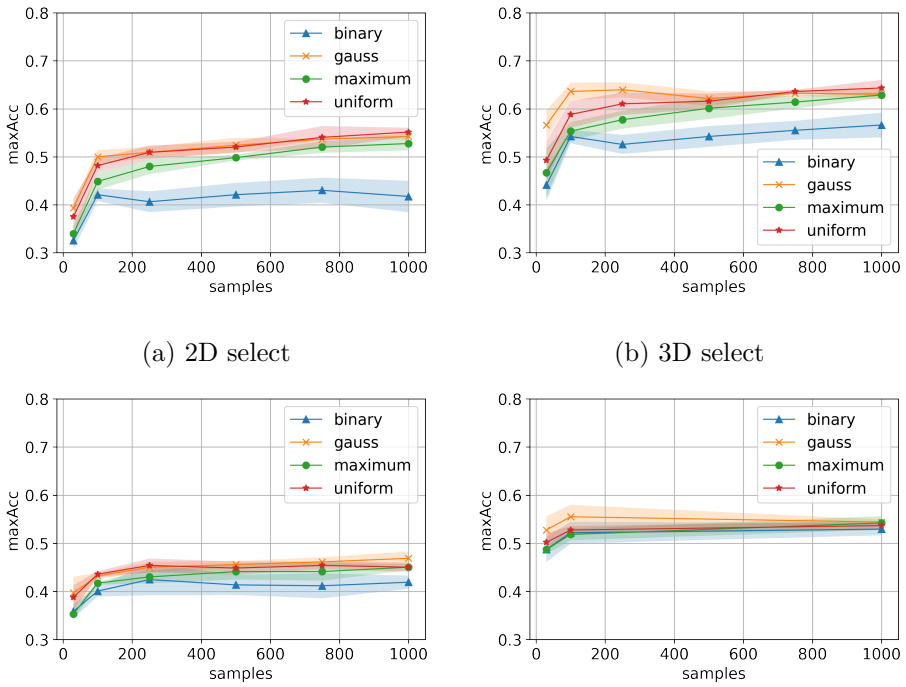
The call-sign-command recovery model (CCR), displayed in Figure 7.2, combines command with call-sign recognition to increase the CRU robustness. It consists of a CallSBERT branch (solid lines) and the additional command branch (dotted lines), which utilizes coordinates as additional input. The command branch consists of three different modules, the command classifier, the command distribution module (CDM) and the final call-sign identifier. The command classifier is a transformer-based multilabel classifier. It can detect whether a transcript contains one or multiple of the six command types described in Section 7.3. The predicted command types are fed into the command distribution module (CDM). The CDM consists of plane 2D/3D-coordinates → command probabilities (*Dis*) mappings for each of the six command types. The CDM contains mappings for each command type and they are selected based on the command types that are recognized by the command classifier. Therefore, the *Dis* scores indicate which plane in



(a) Plane coordinates (b) Binary filtered (c) Gaussian filtered

Figure 7.3: 2D coordinates of airplanes while receiving a vertical command (a) and 2D distribution maps (top view) of the vertical command in the 200 km · 200 km Prague airspace (b),(d). Dark colored areas have a high probability for vertical commands.

the airspace is most likely mentioned in the transcript based on its position and the command uttered in the transcript. In the example in Figure 7.2, just the *horizontal* command type is identified, therefore the CDM only uses the probability distribution map of the *horizontal* command for the *Dis* generation. If there is no transcript available, the coordinate → probability mappings for every command type are considered and mean pooled. For generating the mappings, a small set of coordinate-command pairs of the target airspace are filtered by one of the following filter functions: *Gaussian*, *binary*, *maximum* or *uniform*. The filtering, described in Section 7.4.4 allows one to generate command probability distributions for the whole airspace out of just a few hundred samples as Figure 7.3 shows. The final call-sign identifier of the CCR model takes the *Sim* scores of CallSBERT and the *Dis* scores of the CDM module for each surveillance call-sign and generates a final weighted score for each surveillance call-sign and extracts the most probable one. Our identifier consists of a fully connected five-layer network with relu activations and batch normalization between the fully connected layers and a sigmoid activation at the last layer.



(a) 2D select

(b) 3D select

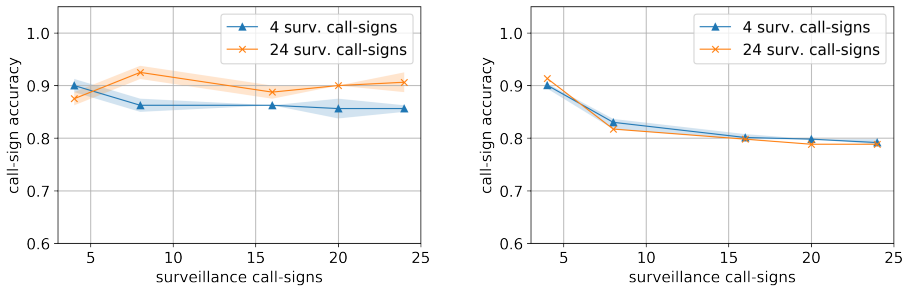
(c) 2D naive

(d) 3D naive

Figure 7.4: Maximum accuracy of call-sign prediction based on command distributions with optimal filter parameters.

7.4.4 CDM optimization

To reduce the need for a large transcribed corpus to create the command probability distributions of the CDM for a new airspace, we evaluate different filter functions for the distribution generation in a low-resource scenario. The naive baseline uses all command distributions for the call-sign prediction (*naive mode*). Using only the relevant command distributions (*select mode*), which are selected by the command classifier of the CCR, adds 10% accuracy to the naive baseline for the *Gaussian*, *maximum* and *uniform* filter as Figure 7.4 shows. Switching from 2D coordinates to 3D coordinates, respectively incorporating the plane height, additionally adds 10% accuracy. The highest accuracy for the low resource scenario is achieved with a Gaussian filter. Using just 100 coordinate-command pairs to generate the distributions via Gaussian filtering gives a similar accuracy as with 1000 samples for the *3D select* case. For the final CCR model, we therefore use



(a) EncDec (b) CallSBERT

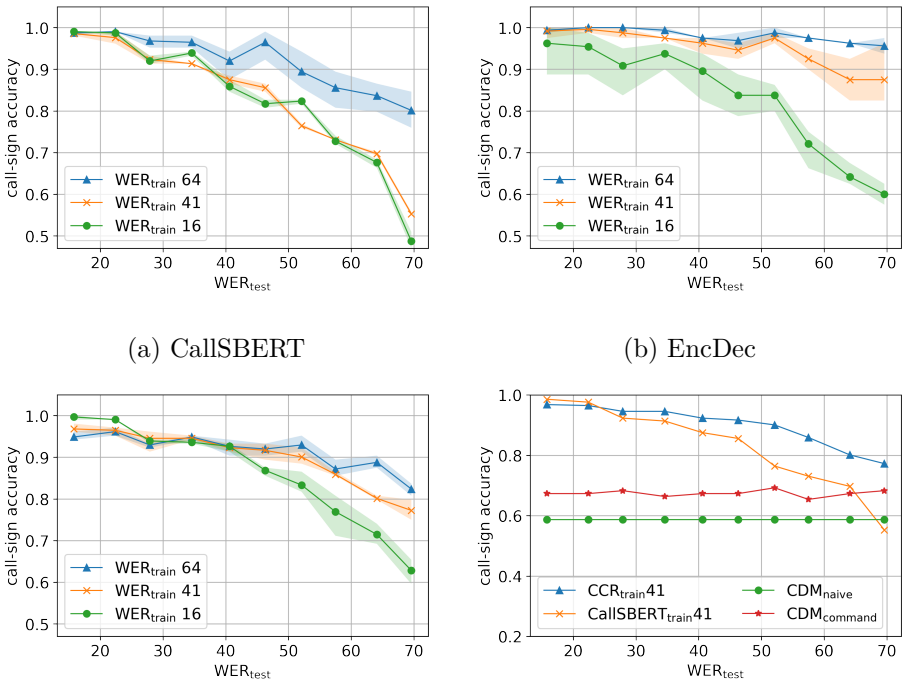
Figure 7.5: Call-sign accuracy depending on the surveillance size per test transcript. During fine-tuning, each transcript has either 4 or 24 corresponding surveillance call-signs.

3D coordinate→probability mappings generated by a Gaussian filtering as shown in Figure 7.3c.

7.5 Results

7.5.1 CallSBERT: Surveillance adaptation

Depending on the flight sector, the amount of surveillance call-signs available might vary. The EncDec architecture has been proven to be robust against fluctuations in the count of surveillance call signs during testing (Blatt, Kocour, et al., 2022). The question remains how the EncDec architecture and CallSBERT react, when they are fine-tuned with a different amount of surveillance call-signs. Figure 7.5 shows that the CA of the EncDec model, despite staying over 80%, depends on the number of surveillance call-signs encountered during training. If the model is finetuned on samples with 24 surveillance call-signs per transcript, it performs better if the number of surveillance call-signs during testing is in the same range. The same holds true for the model trained with 4 surveillance call-signs per transcript. The CallSBERT model, however, seems to be agnostic against the number of surveillance call-signs encountered during training and shows the



(a) CallSBERT (b) EncDec

(c) CCR (d) CCR ablation study

Figure 7.6: Call-sign accuracy depending on the WER of the MALORCA test data.

expected behaviour of a reduced CA with an increasing number of surveillance call-signs due to an increasing search-space.

7.5.2 Edge cases

7.5.2.1 High word error rate

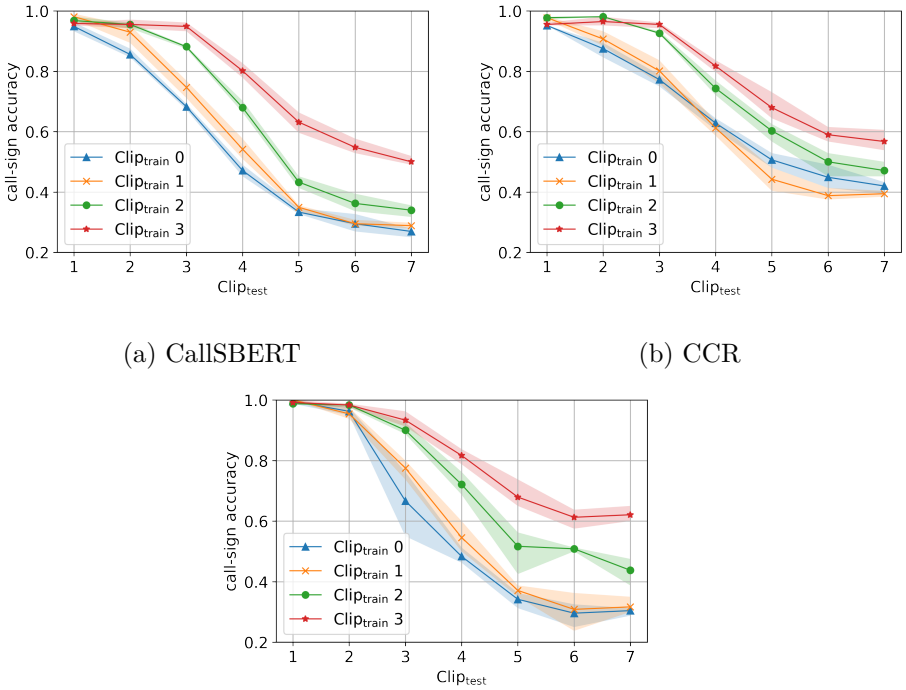
The best performing SOTA ASR model of (Kocour, Veselý, Szöke, et al., 2022) achieves a mean word error rate (WER) on their LiveATC data set (Kocour, Veselý, Szöke, et al., 2022) of 26.8%. But we found that 24% of the transcripts have a 40% WER or higher and 9% of the transcripts have even a WER over 60%. For our experiments, we therefore generate test data sets with a mean WER of up to 70%. Figure 7.6 shows that both, the EncDec and CallSBERT models show a significant performance drop at high WERs, when trained on low WER data of 16%. Training on data with higher WERs allows the models to learn the noise

distribution and reduces CA deterioration by up to 30%, with the larger EncDec model adapting better to ASR noise. Incorporating the CallSBERT model into the CCR architecture stabilizes the CA for a WER over 60% and adds up to 15% to the accuracy of the pure CallSBERT model.

To further evaluate this, we conduct an ablation study on the CCR architecture. In the $\text{CDM}_{\text{naive}}$ case, the CDM mean pools the output of all 3D command distributions to generate a score for each plane coordinate. Since this part of the CCR is not depending on ASR output, the accuracy is stable over the whole WER range as Figure 7.6d shows. By feeding the output of the command classifier into the CDM ($\text{CDM}_{\text{command}}$), the CDM selects the distribution map of the most probable command for predicting the call-sign. This adds roughly 10% performance as Figure 7.6d shows. The missing deterioration of the accuracy at high WERs proves the robustness of the command prediction. Up to a test WER of 40%, the call-sign accuracy of CallSBERTs is more than 20% higher than the CA of the $\text{CDM}_{\text{command}}$. At higher WERs the accuracy of CallSBERT drops significantly. The full CCR architecture however outperforms the single CCR modules by combining the $\text{CDM}_{\text{command}}$ and CallSBERTs output. The ablation study highlights the importance of using multi-modal data, but also the importance of extracting noise-robust text-based features.

7.5.2.2 Clipping

An ATC utterance can be clipped at the beginning if the transmission of the utterance starts delayed after the ATCO or pilot starts talking. CRU algorithms are quite sensitive to clipping, since the call-sign is either located at the beginning or end of an utterance. Clipping for example the first three words of the call-signs `lufthansa one two four lima echo` and `ryanair three five four lima echo` results in the identical call-sign. Figure 7.7 shows that clipping just the first four words reduces the CA of CallSBERT below 50%, which is comparable to a WER higher than 70%. With the clipping of six words, the CA starts to plateau, since the majority of call-signs at the beginning of utterances are cut off beyond



(a) CallSBERT

(b) CCR

(c) EncDec

Figure 7.7: Call-sign accuracy depending on the number of words clipped off at the beginning of the transcripts.

recognition. Training specifically on those shortened utterances can recover up to 30% CA. The additional command branch of the CCR reduces the performance drop by 10%, even for a CallSBERT model, which is trained on unclipped data. The comparison between the CCR module and the EncDec architecture shows that both architectures have a similar performance, when they encounter already heavily clipped data during training. If however only one or no words are clipped during training the CCR architecture outperforms the EncDec model significantly.

7.5.2.3 Missing transcript

The worst-case scenario for a CRU model is a missing transcript. In the ATCO^2 project, utterances with an SNR < 0 dB, make up roughly 10% of all the recordings. They are however discarded because they are too noisy for ASR. To still make use of such samples, a CRU model has to work solely on surveillance data. Table 7.1

Table 7.1: Call-sign accuracy on test data without transcripts.

WER_{train}	16%	41%	64%
CallSBERT	0.03(±0.02)	0.07(±0.06)	0.06(±0.04)
EncDec	0.00(±0.00)	0.12(±0.04)	0.31(±0.05)
CCR	**0.16**(±0.04)	**0.33(±0.03)**	**0.37**(±0.04)

shows, that the CallSBERT model cannot utilize the surveillance call-signs to reach a CA higher than 10% if the transcript is completely missing. The EncDec model is capable of generating predictions, when trained on the 64% WER data because the model utilizes the simultaneous processing of all surveillance call-signs to draw a prediction from previous surveillance constellations. It falls however far behind the CCR model for lower WER training data and fails completely at 16% WER training data. The additional command distribution maps keeps the CCR module still operational at 16% WER, where the other CRU models completely break down.

7.6 Conclusion

In this work we have shown at the example of call-sign recognition and understanding models, that edge-case optimization leads to a more stable performance over a broad operational range. Fine-tuning on noisy transcripts reduces the noise-introduced accuracy drop significantly without degrading accuracy levels on clean data. This holds true for high-WER transcripts and for word-clipped transcripts. Our introduced CallSBERT model shows just a minor performance decrease compared to the EncDec model introduced in (Blatt, Kocour, et al., 2022) while having only 37.1% of the parameters and being faster and more robust during fine-tuning. This performance gap is significantly reduced when CallSBERT is integrated in our newly proposed multimodal CCR architecture. The

ablation study of the architecture shows that the additional context information extracted by the command distribution module and the command classification module of the CCR architecture ensures a stable performance for all investigated edge-case scenarios. This makes this design also interesting for other domains, where coordinates of communication targets are known, for example the nautical or the military domain. Due to its command distribution module, the CCR model can even produce nearly 40% accurate predictions when there is no transcript available, making it the favorable choice for a robust call-sign prediction model.

8

Automatic Readback Error Detection for Air-traffic Control

Contents

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 demand for automated 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 in an unseen target airspace. In this chapter, 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.

The content of this chapter is based on:

> Bashyam, Lakshmi Rajendram, **Blatt, Alexander**, and Dietrich Klakow (2023). "Enabling Noisy Label Usage for Out-of-Airspace Data in Read-Back Error Detection." In: *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1–8. DOI: 10.1109/ASRU57964.2023.10389759.

8.1 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. (Degas et al., 2022) 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 (European Union Aviation Safety Agency, 2021). Assistant tools like auto-pilot or arrival manager (Ohneiser, Helmke, et al., 2021) 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 (Cardosi et al., 1998). With air-traffic control (ATC) being responsible for 6-10% of aircraft crashes (Nikšić et al., 2022), 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 (Zuluaga-Gomez, Veselý, Szöke, et al., 2022). 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 (Kocour, Veselý, Szöke, et al., 2022) and a tool for recognizing call-signs in noisy air-traffic transcripts by using surveillance information (Blatt, Kocour, et al., 2022). 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 call-sign of the addressed plane, in the example, this would be `LUF674F`. The call-sign 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 call-sign of the plane is also read back in most of these cases.

In longer conversations, the read-back can also miss the call-sign or include abbreviated versions of the call-sign as stated by Blatt et al. (Blatt, Kocour, et al., 2022), which complicates read-back error classification. To further complicate the matter, read-back errors occur just in 1-4% of the uttered commands (Cardosis, 1994; Prinzo et al., 2009; Helmke, Kleinert, Shetty, et al., 2021). 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-

1 MALORCA Homepage: https://www.malorca-project.de/
2 ATCO2 Homepage: https://www.atco2.org/

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 (Chen et al., 2017; Cheng et al., 2018; JIA et al., 2018b; Helmke, Kleinert, Shetty, et al., 2021; Helmke, Ondřej, et al., 2022).

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.

8.2 Related work

Because of the severe consequences, the causes of ATC errors are the target of several studies. Marrow et al. (Morrow et al., 1993) identify amongst others the length of an ATC message and the amount of traffic as causes for errors in ATC communications. Cardosi et al. (Cardosi et al., 1998) uncover wrong pilot expectations, pilots sharing the same frequency and a high controller workload as additional factors. In a more recent work by Wu et al. (Wu et al., 2019) 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. (Chen et al., 2017). They propose an automatic speech recognition (ASR) based system, that features a GUI to display read-back error alerts. Jia et al. (JIA et al., 2018b) 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. (Helmke, Ondřej, et al., 2022). In the first step, the ATC transcripts are converted, either rule-based or transformer-based (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Aidan N Gomez, et al., 2017a), 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-based approach (Devlin, M. Chang, et al., 2019) is used for read-back error detection.

However, 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 the generation of noisy labels. We especially build on two previous noisy label works. Firstly, Zhu et al. (D. Zhu et al., 2022) have shown that noisy labels can be used with BERT without using advanced noise handling methods, such as noise matrices. Secondly, Goh et al. (Goh et al., 2018) 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.

8.3 Methods

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

8.3.1 Read-back Error Classes

In the scope of this paper, we focus on pairwise 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-

Table 8.1: 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

pilot utterance pairs for each class are given in Table 8.1.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 than 10% of the `Wrong pair` samples.

8.3.2 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 (Godfrey, 1994) consists of ATC communication and transcripts from 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 (Kocour, Veselý, Szöke, et al., 2022), consists of ATC radio transcriptions, 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 952 samples is built by manually categorizing these pairs into the classes listed in Table 8.1. The manual labeling is performed by the author and supported by experts of the ATC domain to ensure proper labeling and the selection of appropriate 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 (Holub et al., 2008) to select 20 additional samples from the unlabeled pool. This cycle is repeated, with the training pool growing with each iteration, until a total of 317 additional samples are acquired. This increases the likelihood of discovering informative sample pairs within the data pool. After active learning, the sample pool consists of 1232 samples as Table 8.2 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 8.1. As already discussed in previous works, the distribution is unbalanced, with the `Correct class` making up more than

3 LiveATC website: https://www.liveatc.net/

Table 8.2: Class distribution of samples in the initial pool, collected by active learning (AL), data augmentation (Aug), and rule based system (Noisy)

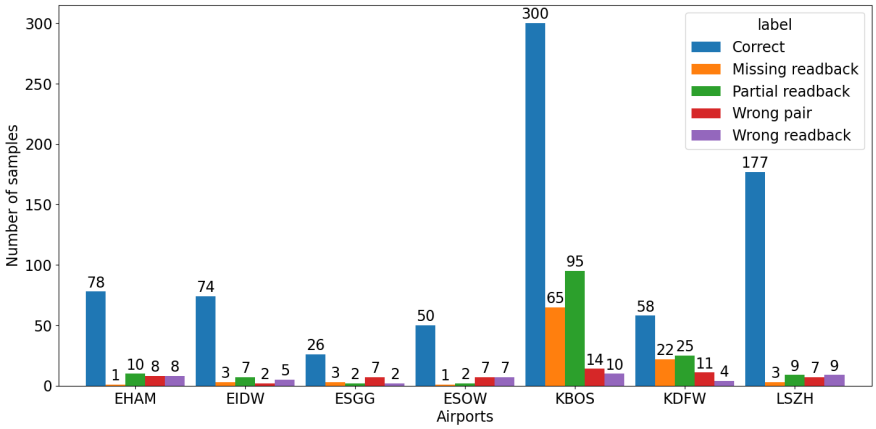
Method	Partial	Missing	Correct	Wrong	Wrong pair	Total
Initial	93	66	712	41	40	952
AL	84	58	138	6	31	317
Aug	763	499	0	514	0	1776
Noisy	1188	1143	4469	1898	1407	10105

60% of the samples. One way to address this is to perform data augmentation as described in Section 8.3.4

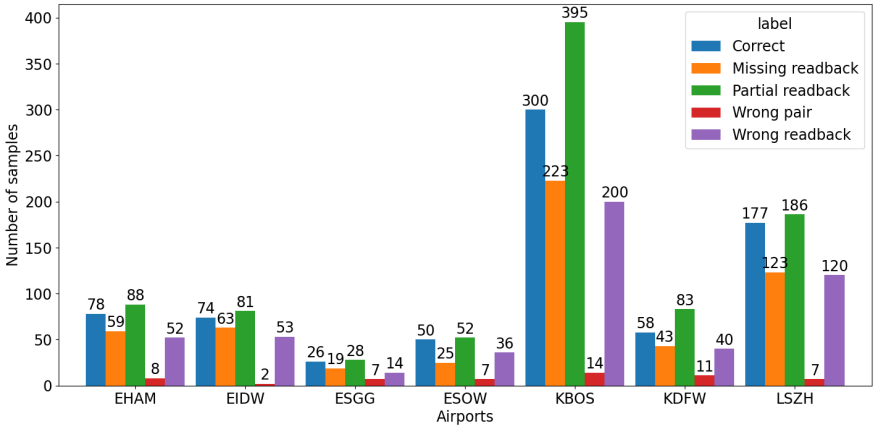
8.3.3 Number Standardization

A closer look at Table 8.1 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 8.1 for example, the pilot and command utterance contain the same value, `132755`. To distinguish between the `Wrong Pair` class and the other classes, the comparison of the call-signs in ATCO and pilot transcripts is equally important. Since command values and call-signs 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 matching difficult. To overcome this issue, we format each number in a standard format by splitting it into its individual digits using a tool provided by Brandhsu et al. (Brandhsu, 2022).



(a) Label distribution across the different airport airspaces.



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

Figure 8.1: Label distribution before (a) and after (b) data augmentation.

8.3.4 Data Augmentation

To address the low occurrence of read-back error cases, we augment 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 call-sign 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 8.1(b) shows the label distribution of each airport airspace after the augmentation. In comparison with Figure 8.1(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.

8.3.5 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 unlabeled 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 in the form of 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 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 call-sign 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 call-signs.

4. `Wrong pair` read-back classes occur when a different pilot than expected responds to the ATCO command (see Table 8.1) or when a new pilot starts communicating on a specific frequency. The missing call-sign 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.

8.4 Experimental Setup

We are investigating the scenario, where the read-back error system is only tested on unseen airspaces, to examine the inter-airspace transferability of the different algorithms. We do not generate augmented data for the test airspace to ensure a realistic test scenario. Figure 8.1 shows, that without augmentation, there exist just a few samples for the majority of the read-back error classes per airport. For

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

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. (Helmke, Ondřej, et al., 2022). 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 8.3, 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 Section 8.3.4. The fifth method uses the noisy labeled data, described in Section 8.3.5. Zhu et al. (D. Zhu et al., 2022) 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. (Goh et al., 2018) 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 exists 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. (Goh et al., 2018) to avoid overfitting.

In the sixth method, we make use of all the available datasets, including manually annotated, augmented and noisy data, in a two-stage noisy + aug-

Table 8.3: 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)

mented training. In this approach, the noisy labels are used to initially fine-tune a pretrained 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.

8.5 Results

We show in the following the results of training our RED system with the six different methods explained in Section 8.4 to evaluate the inter-airspace transferability of the methods. Table 8.3 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

Table 8.4: 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)**

over 15%. Even the second-best algorithm, the two-stage noisy approach, still has 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 8.4. 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. (Helmke, Ondřej, et al., 2022). Their solely machine learning-based system in (Helmke, Ondřej, et al., 2022) 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 better understand 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 8.2. 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 method performs already quite well

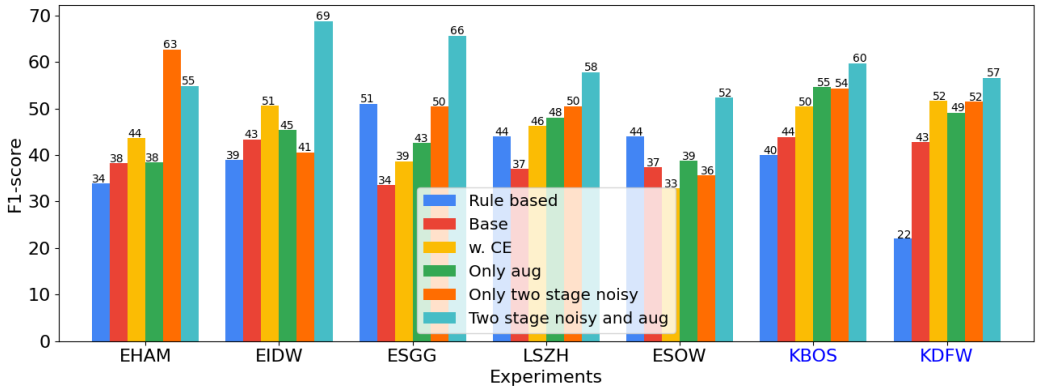


Figure 8.2: F1 scores for the individual airport airspaces. The American airports are marked in blue

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 8.1. 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 labels 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.

8.6 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 also be used in other low-resource domains, where there is a domain mismatch between noisy labels and the test data.

8.7 Future Work

Initial experiments with other evaluation metrics for imbalanced datasets, like Focal loss (A. Gupta et al., 2020) 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.

9

Summary and Outlook

9.1 Summary of contributions

This thesis contributes to several parts of the ATC speech processing pipeline. Starting from the beginning of the pipeline, an analysis is performed if anonymization harms stress detection for pilots (Chapter 3). Going further in the pipeline, we investigate how lexical and acoustic differences between ATC datasets influence the performance of transformer-based ASR models (Chapter 4). Following on from that, a comparison is drawn between cascading ASR and SRD in the pipeline, by using a joint ASR and SRD model (Chapter 5). After the ASR step, text-based models build the end of the speech processing pipeline. For one of the most crucial ATC task, call-sign recognition, two robust algorithms are proposed, which utilize surveillance call-signs (Chapter 6) and additionally plane coordinates (Chapter 7) as input for a more robust recognition. Another crucial task, read-back error detection, is also investigated in terms of robustness and adaptation to unseen airspaces (Chapter 8). In the following, we discuss the main contributions of this thesis.

Robust & generalizing ATC-NLP - Our main focus lies on robustness and generalization. There are two main challenges for any new algorithm developed for ATC communication. The first challenge is the quality of speech, while highly distorted speech is mostly sorted out in public training datasets, it cannot be

sorted out in operation. The second challenge is that publicly available training data covers only a small fraction of the global airspaces. Our contributions to the first challenge target the task of call-sign recognition and understanding. In Chapter 6, we show that injecting ASR noise in the transcripts improves the performance of the CRU model significantly. Our proposal to additionally append surveillance call-signs to the transcript leads to a performance increase that is even multiple times higher. A rigorous analysis shows that this performance increase is robust against unfavorable combination of surveillance call-signs, specifically very similar call-signs or a large quantity of call-signs. Building on this work, we analyze in Chapter 7 the most dominant edge cases, that an CRU system can encounter, high WER transcripts, clipped transcripts and the complete loss of the transcript. We show that the adaptation of a CRU model to the edge cases not only increases its robustness, but also does not harm its performance on normal transcripts. We also take it a step further from only including surveillance call-signs, to additionally including surveillance coordinates. Our results show that this further robustifies the predictions at edge-cases. It needs however an architecture change to combine the different modalities.

To reduce the influence of the low airspace coverage, we propose in Chapter 6 a data augmentation pipeline for CRU which allows to produce realistic transcripts by utilizing publicly available global surveillance call-sign lists. In Chapter 8 we go a step further and explore how data augmentation and noisy labeling can be combined at the example of read-back error detection. We can show that a two-stage fine-tuning approach, first on noisy, then on clean and augmented data outperforms other strategies on unseen airspaces. We furthermore introduce, to the best of our knowledge, the first fully ML based RED system that is able to distinguish between multiple read-back error classes. A task that is challenging due to the highly imbalanced training data.

In summary, we propose several approaches that make NLP tasks in ATC more robust and provide crucial insight into the importance of edge-case robustification.

In combination with our proposed methods to overcome the low data availability, we have paved the way for reliable ML-based assistance systems in ATC.

Understanding and improving ASR in ATC - Our secondary focus lies on ASR. Since NLP algorithms for ATC are significantly influenced by the quality of the ASR output, ASR is one of the most important steps in an ATC speech-processing pipeline. With the upcoming of pretrained transformer-based ASR models like whisper or XLSR, there have been attempts to utilize them also for ATC. They show however a poor generalization behaviour when applied to unseen ATC datasets. Our investigations show that lexical and acoustic differences between the datasets both influence the ASR system. Based on our results, we suggest how the ASR performance on unseen datasets can be predicted by measuring these differences. Our separate noise analysis shows that clean speech combined with Gaussian noise at a certain noise level gives a good estimate of a lower WER bound for other noise types at this noise level. The lexical analysis, on the other hand, demonstrates that dominant OOVs are airspace-dependent cities, greetings and airlines. We identify a target-dataset-specific language model as a way to reduce the influence of both types of differences. Finally, a feature analysis on wav2vec 2.0 itself is performed. The analysis reveals, amongst others, that the wav2vec 2.0 feature encoder is agnostic to lexical changes. We also identify a same-similarity cluster in the transformer encoder which indicates good generalization.

Since the transcription task is in ATC often accompanied by a speaker role detection (SRD) task, which separates the transcripts in `PILOT` and `ATCO` transcripts, we propose a joint transformer-based ASR&SRD model and compare it with traditional cascaded approaches. The joint system is superior in the majority of the tested intra-dataset scenarios, but its performance decreases when applied to new airspaces. We can show however, that few-shot training with a few dozen samples can cure this decrease, making it a compact and efficient alternative to the bigger cascaded architectures.

The result of these investigations is a better understanding of ASR in ATC as well as a better understanding of how ASR and SRD should be combined

depending on the available data. This allows to generate higher quality transcripts for the downstream NLP tasks in the ATC speech-processing pipeline.

Influence on privacy measures speech-based ATC tasks - As discussed above, there is a lack of training data in ATC regarding airspace coverage. There exists however a lot of private speech data, for example recorded by ANSPs. Unfortunately, they often can not share this data due to data protection regularization. We therefore investigate in Chapter 3 whether speaker anonymization techniques can be used to remove the speaker footprint from ATC voice data without harming downstream tasks in the speech processing pipeline. We analyze this at the example of stress detection for ATCOs since on one hand, this is an important task to reduce incidences and on the other hand, it is a task that relies on acoustic features, which are influenced by anonymization. Our results show consistently that anonymization not only does not harm stress detection, but also works as a data augmentation method and improves the results in the cross-domain scenario. These results open the doors for ANSPs to make their data publicly available, which would lead to a more robust database for training ML support systems for ATC.

9.2 Future work

In the second part of this chapter, we discuss potential targets for future work in the context of the ATC speech processing pipeline.

Explainability - Large language models like GPT-4 are widely used in chatbots, and are capable of producing elaborated answers to complex questions. The problem is however, that they often hallucinate and produce wrong answers that seem plausible. Especially for a safety critical industry like ATC, relying on LLMs is a huge risk. There are however promising approaches that try to minimize hallucinations in other domains such as the medical domain (Ji, Yu, et al., 2023).

More research is needed to adapt these algorithms to the ATC domain. Another way to mitigate errors is to correctly estimate the uncertainty of the model, which allows one to cope with over-and under-confidence (Gawlikowski et al., 2023). Incorporating these methods in ATC speech processing algorithms would allow us to sort out uncertain predictions, thus resulting in a safer automation.

Robustness and Multimodality - Especially when it comes to proving that a model is ready to be used in operation, it is crucial to show its robustness. We therefore propose the development of a speech and text benchmark dataset for edge-cases in ATC as a first necessary step. An additional edge case to the ones mentioned in this thesis could be low-proficient speakers, since this is still a problem among pilots (Lynch et al., 2021). Another interesting edge case are emergencies, because the structure and vocabulary of emergency messages differ significantly from the standard ATC phraseology, and emergencies are also a part of language proficiency tests for pilots (Petrashchuk et al., 2019).

Equally important to adequate robustness testing is the robustification of the algorithms themselves. We have shown that multimodality is one way to achieve this. Including plane trajectory predictions (Zeng et al., 2022) into read-back error detection would for example give the benefit of detecting potential erroneous flight behaviour due to miscommunication before it leads to incidents. Robustifying ATC ML models for the loss of modalities (McKinzie et al., 2023; Hazarika et al., 2020) should be also taken into account to guarantee a stable operation of the future algorithms.

Task Joining - We have shown that performing ASR and SRD in one step is not only more efficient but can even outperform other two-step approaches. We think that there are also other tasks that would benefit from a combination. The next logical step from combined ASR&SRD would be combined ASR&RED, since read-back error detection relies on differentiating `Pilot` and `ATCO` transcripts. Another interesting task is named-entity recognition (NER) for ATC, with the

entities `call-sign`, `command` and `value`, which is currently performed on transcripts. Since the pilot and ATCO utterances differ in the sentence structure, but also in their acoustic features, NER could benefit from being integrated in the ASR tasks. Current approaches that combine ASR and NER, like WhisperNER (Ayache et al., n.d.), show promising results. It is also worth investigating the combination of multiple tasks with ASR, which would for example result in a transcript that is enriched with SRD, RED and NER tags.

Adaptation and Generalization - While there exists a growing number of works targeting ATC speech processing, there are still many blind spots when it comes to data coverage. Common databases like ATCO2, LDC-ATCC, NATS or ISAVIA cover European and American airspaces, but datasets for whole continents like South America or Africa are, to the best of our knowledge, missing. Furthermore, some of the existing datasets, such as NATS or ISAVIA, are private datasets, which further reduces the general coverage. This poses the risk of performance degradations at those unseen airspaces due to unknown regional way-points, call-signs, accents or languages. There exist approaches to adapt models to a number of unseen languages (Alabi et al., 2022), alternatively zero-shot accent adaptation can be done (Owodunni et al., 2024) and we have presented augmentation methods to adapt to regional call-sign via ADS-B data. Although these methods allow adaptation to some degree, global data collection should not be neglected to achieve a better generalization of the models.

Privacy in ATC - The availability of ATC speech data often depends on country-specific laws. In Germany, it is for example forbidden to record others without asking them for their permission, as stated in § 201 StGB. Such restrictions are based on the right to individual privacy. There exist many works on speech and text anonymization (Panariello et al., 2024; Sousa et al., 2023), but it must be determined which privacy-preserving algorithms are suitable to allow (anonymized) ATC speech recording despite the local legislation. Withing this research, the

special phraseology of ATC speech must be considered to guarantee, that for example pretrained algorithms can fulfill their privacy guarantees but also generate useful data for downstream tasks in the new domain. We have already shown that anonymization can improve generalization and therefore think that research in this direction will not only lead to more, but also to better data.

10

Abbreviations

AAL	American airlines
ADS-B	Automatic Dependent Surveillance–Broadcast
ALTAI	assessment list for trustworthy AI
AMAN	arrival manager
ANN	artificial neural networks
ANSP	air navigation service providers
ASR	automatic speech recognition
ASV	automatic speaker verification
ATC	air-traffic control
ATCO	air-traffic controller
ATM	air-traffic management
BERT	bidirectional encoder representations from transformers
Bi-LSTM	bidirectional long short-term memory
CCR	call-sign-command recovery model
CDM	command distribution module
CNN	convolutional neural network
CPDLC	controller–pilot data link communications
CTC	connectionist temporal classification
CUDA	Compute Unified Device Architecture
DCT	discrete cosine transform
DFS	Deutsche Flugsicherung GmbH
DLH	Deutsche Lufthansa

EASA	European Union Aviation Safety Agency
FAR	false acceptance rate
FFN	feedforward neural network
FRR	false rejection rate
GD	gradient descent
GDPR	general data Protection regulation
GPU	graphics processing unit
IOB	inside,outside, beginning
LMS	log mel spectrogram
LLM	large language model
LSTM	long short-term memory
MFCC	Mel-frequency cepstral coefficients
MLM	masked language modeling
MSE	mean squared error
NER	named-entity recognition
NLP	natural language processing
NLU	natural language understanding
NN	neural network
RED	read-back error detection
RNN	recurrent neural network
SD	speaker diarization
SGD	stochastic gradient descent
SOTA	state-of-the-art
SRD	speaker role detection
StGB	Strafgesetzbuch
STT	speach-to-text
VHF	very high frequency
WER	word error rate

List of Figures

List of Tables

Bibliography

Ackley, David H., Geoffrey E. Hinton, and Terrence J. Sejnowski (1985). "A learning algorithm for boltzmann machines." In: *Cognitive Science* 9.1, pp. 147–169. ISSN: 03640213. DOI: 10.1016/S0364-0213(85)80012-4 (cit. on p. 2).

Adelani, David Ifeoluwa, Ali Davody, Thomas Kleinbauer, Dietrich Klakow, David Ifeoluwa Adelani, Ali Davody, Thomas Kleinbauer, Dietrich Klakow, David Ifeoluwa Adelani, Ali Davody, Thomas Kleinbauer, and Dietrich Klakow (2020). "Privacy guarantees for de-identifying text transformations To cite this version : HAL Id : hal-02907939 Privacy Guarantees for De-identifying Text Transformations." In: (cit. on p. 41).

Alabi, Jesujoba O., David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow (2022). "Adapting Pre-trained Language Models to African Languages via Multilingual Adaptive Fine-Tuning." In: *Proceedings - International Conference on Computational Linguistics, COLING* 29.1, pp. 4336–4349. ISSN: 29512093. arXiv: 2204.06487 (cit. on p. 138).

Alharasees, Omar, Abeer Jazzar, Utku Kale, and Daniel Rohacs (2023). "Aviation communication: the effect of critical factors on the rate of misunderstandings." In: *Aircraft Engineering and Aerospace Technology* 95.3, pp. 379–388. ISSN: 17488842. DOI: 10.1108/AEAT-02-2022-0052 (cit. on p. 19).

Alshammari, Nasser and Saad Alanazi (2021). "The impact of using different annotation schemes on named entity recognition." In: *Egyptian Informatics Journal* 22.3, pp. 295–302. ISSN: 11108665. DOI: 10.1016/j.eij.2020.10.004. URL: https://doi.org/10.1016/j.eij.2020.10.004 (cit. on p. 31).

Amari, Shun-ichi (1993). "Backpropagation and stochastic gradient descent method." In: *Neurocomputing* 5.4, pp. 185–196. ISSN: 0925-2312. DOI: https://doi.org/10.1016/0925-2312(93)90006-O. URL: https://www.sciencedirect.com/science/article/pii/092523129390006O (cit. on p. 22).

Apicella, Andrea, Francesco Donnarumma, Francesco Isgrò, and Roberto Prevete (2021). "A survey on modern trainable activation functions." In: *Neural Networks* 138, pp. 14–32. ISSN: 18792782. DOI: 10.1016/j.neunet.2021.01.026. arXiv: 2005.00817. URL: https://doi.org/10.1016/j.neunet.2021.01.026 (cit. on p. 23).

Aridor, Guy, Yeon-Koo Che, William Nelson, and Tobias Salz (2020). "The Economic Consequences of Data Privacy Regulation: Empirical Evidence from GDPR." In: *SSRN Electronic Journal*. DOI: 10.2139/ssrn.3522845. URL: https://piwik.pro/blog/privacy-laws-around-globe/. (cit. on p. 35).

Aurelio, Yuri Sousa, Gustavo Matheus de Almeida, Cristiano Leite de Castro, and Antonio Padua Braga (2019). "Learning from Imbalanced Data Sets with Weighted Cross-Entropy Function." In: *Neural Processing Letters* 50.2, pp. 1937–1949. ISSN: 1573773X. DOI: 10.1007/s11063-018-09977-1. URL: https://doi.org/10.1007/s11063-018-09977-1 (cit. on p. 37).

Ayache, Gil, Menachem Pirchi, Aviv Navon, Aviv Shamsian, Gill Hetz, and Joseph Keshet (n.d.). "WhisperNER : Unified Open Named Entity and Speech Recognition." In: (). arXiv: arXiv:2409.08107v1 (cit. on p. 138).

Babu, Arun, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli (2022). "XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale." In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH* 2022-September, pp. 2278–2282. ISSN: 19909772. DOI: 10.21437/Interspeech.2022-143. arXiv: 2111.09296 (cit. on pp. 6, 76, 78).

Baevski, Alexei, Henry Zhou, Abdelrahman Mohamed, and Michael Auli (2020). "wav2vec 2.0: A framework for self-supervised learning of speech representations." In: *Advances in Neural Information Processing Systems* 2020-Decem.Figure 1, pp. 1–19. ISSN: 10495258. arXiv: 2006.11477 (cit. on pp. 6, 25, 27, 28, 33, 76, 78).

Basha, S. H.Shabbeer, Shiv Ram Dubey, Viswanath Pulabaigari, and Snehasis Mukherjee (2020). "Impact of fully connected layers on performance of convolutional neural networks for image classification." In: *Neurocomputing* 378, pp. 112–119. ISSN: 18728286. DOI: 10.1016/j.neucom.2019.10.008. arXiv: 1902.02771. URL: https://doi.org/10.1016/j.neucom.2019.10.008 (cit. on pp. 23, 25).

Bathla, Gourav, Kishor Bhadane, Rahul Kumar Singh, Rajneesh Kumar, Rajanikanth Aluvalu, Rajalakshmi Krishnamurthi, Adarsh Kumar, R. N. Thakur, and Shakila Basheer (2022). "Autonomous Vehicles and Intelligent Automation: Applications, Challenges, and Opportunities." In: *Mobile Information Systems* 2022. ISSN: 1875905X. DOI: 10.1155/2022/7632892 (cit. on p. 2).

Bellagha, Mohamed Lazhar and Mounir Zrigui (2020). "Speaker Naming in TV programs Based on Speaker Role Recognition." In: *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA* 2020-November. ISSN: 21615330. DOI: 10.1109/AICCSA50499.2020.9316511 (cit. on p. 34).

Benesty, Jacob, Jingdong Chen, Yiteng (Arden) Huang, and Simon Doclo (Dec. 2005). "Study of the Wiener Filter for Noise Reduction." In: *Speech Enhancement*. Springer-Verlag, pp. 9–41. DOI: 10.1007/3-540-27489-8_2. URL: https://link.springer.com/chapter/10.1007/3-540-27489-8%7B%5C_%7D2 (cit. on p. 46).

Blatt, Alexander, Badr M. Abdullah, and Dietrich Klakow (2023). "Ending the Blind Flight: Analyzing the Impact of Acoustic and Lexical Factors on WAV2VEC 2.0 in Air-Traffic Control." In: *2023 IEEE Automatic Speech Recogni-*

tion and Understanding Workshop (ASRU), pp. 1–8. DOI: 10.1109/ASRU57964.2023.10389646 (cit. on pp. 75, 76, 81).

Blatt, Alexander, Martin Kocour, Karel Veselý, Igor Szöke, and Dietrich Klakow (2022). "Call-Sign Recognition and Understanding for Noisy Air-Traffic Transcripts Using Surveillance Information." In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* 2022-May.864702, pp. 8357–8361. ISSN: 15206149. DOI: 10.1109/ICASSP43922.2022.9746301 (cit. on pp. 101–104, 108, 112, 117, 148).

Borghini, Gianluca, Gianluca Di Flumeri, Pietro Aricò, Nicolina Sciaraffa, Stefano Bonelli, Martina Ragosta, Paola Tomasello, Fabrice Drogoul, Uğur Turhan, Birsen Acikel, Ali Ozan, Jean Paul Imbert, Géraud Granger, Railane Benhacene, and Fabio Babiloni (May 2020). "A multimodal and signals fusion approach for assessing the impact of stressful events on Air Traffic Controllers." In: *Scientific Reports* 10.1, pp. 1–18. ISSN: 20452322. DOI: 10.1038/s41598-020-65610-z. URL: https://www.nature.com/articles/s41598-020-65610-z (cit. on pp. 5, 42).

Brandhsu (2022). *A simple, deterministic, and extensible approach to inverse text normalization for numbers*. https://github.com/barseghyanartur/itnpy (cit. on p. 122).

Cardosi, Kim, Paul Falzarano, and Sherwin Han (1998). *Pilot-controller communication errors: An analysis of Aviation Safety Reporting System (ASRS) reports* (cit. on pp. 116, 118).

Cardosis, M. (1994). "An Analysis of Tower (Local) Controller - Pilot Voice Communications." In: June. URL: https://rosap.ntl.bts.gov/view/dot/8652 (cit. on p. 117).

Caruana, Rich, Steve Lawrence, and Lee Giles (2001). "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping." In: *Advances in Neural Information Processing Systems*. ISSN: 10495258 (cit. on p. 22).

Chadawan Ittichaichareon, Siwat Suksri and Thaweesak Yingthawornsuk (2012). "Speech Recognition using MFCC." In: *International Conference on Computer Graphics, Simulation and Modeling (ICGSM'2012)*, pp. 135–138 (cit. on p. 46).

Chen, Shuo, Hunter Kopald, Ronald S. Chong, Yuan Jun Wei, and Zachary Levonian (2017). "Read back error detection using automatic speech recognition." In: *12th USA/Europe Air Traffic Management R and D Seminar* (cit. on pp. 3, 118).

Cheng, Fangyuan, Guimin Jia, Jinfeng Yang, and Dan Li (2018). "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)*. Vol. 10996 LNCS, pp. 580–588. ISBN: 9783319979083. DOI: 10.1007/978-3-319-97909-0_62 (cit. on pp. 12, 118).

Choi, Kwanghee and Eun Jung Yeo (2022). "Opening the Black Box of wav2vec Feature Encoder." In: arXiv: 2210.15386. URL: http://arxiv.org/abs/2210.15386 (cit. on p. 58).

Cornell, Samuele, Jee-weon Jung, Shinji Watanabe, and Stefano Squartini (2023). "One model to rule them all ? Towards End-to-End Joint Speaker Diarization and Speech Recognition." In: pp. 2–6. arXiv: 2310.01688. URL: http://arxiv.org/abs/2310.01688 (cit. on pp. 75, 76).

Corver, Sifra Christina, Dana Unger, and Gudela Grote (2016). "Predicting Air Traffic Controller Workload: Trajectory Uncertainty as the Moderator of the Indirect Effect of Traffic Density on Controller Workload Through Traffic Conflict." In: *Human Factors* 58.4. PMID: 27076095, pp. 560–573. DOI: 10.1177/0018720816639418. eprint: https://doi.org/10.1177/0018720816639418. URL: https://doi.org/10.1177/0018720816639418 (cit. on pp. 3, 5).

Costa, Giovanni (1996). *Occupational Stress and Stress Prevention in Air Traffic Control*, p. 43. ISBN: 9221100707. URL: http://www.ilo.org/wcmsp5/groups/

public/---ed%7B%5C_%7Dprotect/---protrav/---safework/documents/publication/wcms%7B%5C_%7D250120.pdf%7B%5C%%7D0Ahttps://www.ilo.org/public/libdoc/ilo/1996/96B09%7B%5C_%7D2%7B%5C_%7Dengl.pdf (cit. on pp. 40, 41).

Dawalatabad, Nauman, Mirco Ravanelli, François Grondin, Jenthe Thienpondt, Brecht Desplanques, and Hwidong Na (2021). "ECAPA-TDNN embeddings for speaker diarization." In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH* 4, pp. 2528–2532. ISSN: 19909772. DOI: 10.21437/Interspeech.2021-941. arXiv: 2104.01466 (cit. on p. 8).

De Simone, Valentina, Valentina Di Pasquale, and Salvatore Miranda (2022). "An overview on the use of AI/ML in Manufacturing MSMEs: solved issues, limits, and challenges." In: *Procedia Computer Science* 217.2022, pp. 1820–1829. ISSN: 18770509. DOI: 10.1016/j.procs.2022.12.382. URL: https://doi.org/10.1016/j.procs.2022.12.382 (cit. on p. 2).

Degas, Augustin, 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 (Jan. 2022). "A Survey on Artificial Intelligence (AI) and eXplainable AI in Air Traffic Management: Current Trends and Development with Future Research Trajectory." In: *Applied Sciences (Switzerland)* 12.3, p. 1295. ISSN: 20763417. DOI: 10.3390/app12031295. URL: https://www.mdpi.com/2076-3417/12/3/1295 (cit. on p. 116).

Delpech, Estelle, Marion Laignelet, Christophe Pimm, Céline Raynal, Michal Trzos, Alexandre Arnold, and Dominique Pronto (2019). "A real-life, French-accented corpus of air traffic control communications." In: *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, pp. 2866–2870.

ISBN: 9791095546009. URL: http://trans.sourceforge.net/ (cit. on pp. 91, 92).

Derouault, Anne Marie and Bernard Merialdo (1986). "Natural Language Modeling for Phoneme-to-Text Transcription." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6, pp. 742–749. ISSN: 01628828. DOI: 10.1109/TPAMI.1986.4767855 (cit. on p. 32).

Devlin, Jacob, Ming Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "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*. Vol. 1, pp. 4171–4186. ISBN: 9781950737130. arXiv: 1810.04805. URL: https://github.com/tensorflow/tensor2tensor (cit. on p. 2).

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (Oct. 2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In: arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805 (cit. on pp. 12, 23).

– (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/V1/N19-1423. URL: https://doi.org/10.18653/v1/n19-1423 (cit. on pp. 80, 90, 102, 118).

Ding, Bosheng, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Tuan Luu, and Shafiq Joty (2024). "Data Augmentation using LLMs: Data Perspectives, Learning Paradigms and Chal-

lenges." In: arXiv: 2403.02990. URL: http://arxiv.org/abs/2403.02990 (cit. on p. 36).

Dubey, Shiv Ram, Satish Kumar Singh, and Bidyut Baran Chaudhuri (2022). "Activation functions in deep learning: A comprehensive survey and benchmark." In: *Neurocomputing* 503, pp. 92–108. ISSN: 18728286. DOI: 10.1016/j.neucom.2022.06.111. arXiv: 2109.14545. URL: https://doi.org/10.1016/j.neucom.2022.06.111 (cit. on p. 23).

Ďurčo, Stanislav, Jozef Sabo, Róbert Rozenberg, and Žaneta Miženková (2017). "Means of Cpdlc Using With Atc Procedures in Terminal Maneuvering Area." In: pp. 48–53 (cit. on p. 17).

Eskilsson, Sofie, Hanna Gustafsson, Suleman Khan, and Andrei Gurtov (Sept. 2020). "Demonstrating ADS-B and CPDLC Attacks with Software-Defined Radio." In: *Integrated Communications, Navigation and Surveillance Conference, ICNS* 2020-Septe, pp. 1–9. ISSN: 21554951. DOI: 10.1109/ICNS50378.2020.9222945 (cit. on p. 88).

European Union Aviation Safety Agency (2021). "EASA Concept Paper: First usable guidance for Level 1 machine learning applications." In: 1, pp. 1–174. URL: https://www.easa.europa.eu/newsroom-and-events/news/easa-releases-its-concept-paper-first-usable-guidance-level-1-machine-0 (cit. on pp. 4, 10, 100, 116).

Flemotomos, Nikolaos, Panayiotis Georgiou, and Shrikanth Narayanan (2020). "Linguistically Aided Speaker Diarization Using Speaker Role Information." In: pp. 117–124. DOI: 10.21437/odyssey.2020-17. arXiv: 1911.07994 (cit. on p. 34).

Frank, Rosenblatt (1958). "The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain." In: *Psychological Review* 65.6, pp. 386–408 (cit. on p. 1).

Franzreb, Carlos, Tim Polzehl, and Sebastian Möller (2023). "A Comprehensive Evaluation Framework for Speaker Anonymization Systems." In: August, pp. 65–72. DOI: 10.21437/spsc.2023-11 (cit. on p. 36).

Fredriksson, Teodor, David Issa Mattos, Jan Bosch, and Helena Holmström Olsson (2020). "Data Labeling: An Empirical Investigation into Industrial Challenges and Mitigation Strategies." In: *Product-Focused Software Process Improvement*. Ed. by Maurizio Morisio, Marco Torchiano, and Andreas Jedlitschka. Cham: Springer International Publishing, pp. 202–216. ISBN: 978-3-030-64148-1 (cit. on p. 36).

Gawlikowski, Jakob, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, Muhammad Shahzad, Wen Yang, Richard Bamler, and Xiao Xiang Zhu (2023). "A survey of uncertainty in deep neural networks." In: *Artificial Intelligence Review* 56.s1, pp. 1513–1589. ISSN: 15737462. DOI: 10.1007/s10462-023-10562-9. arXiv: 2107.03342. URL: https://doi.org/10.1007/s10462-023-10562-9 (cit. on p. 137).

Godfrey, John J. (1994). *Air Traffic Control Complete*. DOI: https://doi.org/10.35111/2bg6-nn53. URL: https://catalog.ldc.upenn.edu/LDC94S14A (cit. on pp. 76, 121).

Goh, Garrett B., Charles Siegel, Abhinav Vishnu, and Nathan Hodas (2018). "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*. KDD '18. London, United Kingdom: Association for Computing Machinery, pp. 302–310. ISBN: 9781450355520. DOI: 10.1145/3219819.3219838. URL: https://doi.org/10.1145/3219819.3219838 (cit. on pp. 119, 126).

Graves, Alex, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber (2006a). "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks." In: *ACM International Conference Pro-*

ceeding Series 148, pp. 369–376. DOI: 10.1145/1143844.1143891 (cit. on pp. 28, 29).

Graves, Alex, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber (2006b). "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks." In: *Proceedings of the 23rd international conference on Machine learning*, pp. 369–376 (cit. on p. 76).

Graves, Alex and Navdeep Jaitly (2014). "Towards end-to-end speech recognition with recurrent neural networks." In: *31st International Conference on Machine Learning, ICML 2014* 5, pp. 3771–3779 (cit. on p. 33).

Greff, Klaus, Rupesh K. Srivastava, Jan Koutnik, Bas R. Steunebrink, and Jurgen Schmidhuber (2017). "LSTM: A Search Space Odyssey." In: *IEEE Transactions on Neural Networks and Learning Systems* 28.10, pp. 2222–2232. ISSN: 21622388. DOI: 10.1109/TNNLS.2016.2582924. arXiv: 1503.04069 (cit. on p. 25).

Gregor, Joseph (2001). *Specialists' Report (on American Airlines Flight 77)*. URL: https://www.ntsb.gov/about/Documents/ATC%7B%5C_%7DReport%7B%5C_%7DAA77.pdf (cit. on pp. 18, 145).

Grimaldi, Vincent, Gilles Courtois, and Hervé Lissek (2018). "Objective evaluation of static beamforming on the quality of speech in noise." In: c, pp. 369–374 (cit. on pp. 61, 62, 146).

Guo, Dongyue, Zichen Zhang, Peng Fan, Jianwei Zhang, and Bo Yang (Nov. 2021). "A context-aware language model to improve the speech recognition in air traffic control." In: *Aerospace* 8.11. ISSN: 22264310. DOI: 10.3390/aerospace8110348 (cit. on p. 102).

Gupta, Akhilesh, Nesime Tatbul, Ryan Marcus, Shengtian Zhou, Insup Lee, and Justin Gottschlich (2020). "Class-Weighted Evaluation Metrics for Imbalanced Data Classification." In: DOI: 10.48550/ARXIV.2010.05995. arXiv: 2010.05995. URL: https://arxiv.org/abs/2010.05995%20http://arxiv.org/abs/2010.05995 (cit. on p. 131).

Gupta, Vishwa, Lise Rebout, Gilles Boulianne, Pierre André Ménard, and Jahangir Alam (Sept. 2019). "CRIM's Speech Transcription and Call Sign Detection System for the ATC Airbus Challenge task." In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. Vol. 2019-Septe. International Speech Communication Association, pp. 3018–3022. DOI: 10.21437/Interspeech.2019-1131 (cit. on pp. 9, 90, 101).

Hagmüller, Martin, Erhard Rank, and Gernot Kubin (2006). "Evaluation of the Human Voice for Indications of Workload-induced Stress in the Aviation Environment." In: EEC Note 2006-18. URL: http://www.eurocontrol.int/eec/public/standard%7B%5C_%7Dpage/DOC%7B%5C_%7DReport%7B%5C_%7D2006%7B%5C_%7D023.html (cit. on p. 41).

Hansen, John H.L. and Sanjay Patil (2007). "Speech under stress: Analysis, modeling and recognition." In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4343 LNAI, pp. 108–137. ISSN: 16113349. DOI: 10.1007/978-3-540-74200-5_6. URL: https://link.springer.com/chapter/10.1007/978-3-540-74200-5%7B%5C_%7D6 (cit. on p. 41).

Hansen, John HL, Sahar E Bou-Ghazale, Ruhi Sarikaya, and Bryan Pellom (1997). "Getting Started with SUSAS: A Speech Under Simulated and Actual Stress Database." In: *Eurospeech*, pp. 1743–46. URL: http:::www.ee.duke.eduuResearchhSpeech (cit. on p. 43).

Hasan, Md. Rakibul, Md. Mahbub Hasan, and Md Zakir Hossain (2021). "How many Mel-frequency cepstral coefficients to be utilized in speech recognition? A study with the Bengali language." In: *The Journal of Engineering* 2021.12, pp. 817–827. ISSN: 2051-3305. DOI: 10.1049/tje2.12082 (cit. on p. 33).

Hazarika, Devamanyu and Yingting Li (2020). "Analyzing Modality Robustness in Multimodal Sentiment Analysis." In: arXiv: arXiv:2205.15465v1 (cit. on p. 137).

Helmke, Hartmut, Matthias Kleinert, Nils Ahrenhold, Heiko Ehr, Thorsten Mühlhausen, Oliver Ohneiser, Lucas Klamert, Petr Motlicek, Amrutha Prasad, Juan Zuluaga-Gomez, Jelena Dokic, and Ella Pinska Chauvin (2023). "Automatic Speech Recognition and Understanding for Radar Label Maintenance Support Increases Safety and Reduces Air Traffic Controllers' Workload." In: *Fifteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2023)*. URL: https://elib.dlr.de/196465/ (cit. on p. 3).

Helmke, Hartmut, Matthias Kleinert, Shruthi Shetty, Oliver Ohneiser, Heiko Ehr, Hörður Arilíusson, Teodor S Simiganoschi, Amrutha Prasad, Petr Motlicek, Karel Veselý, Karel Ondrej, Pavel Smrz, Julia Harfmann, and Christian Windisch (2021). "Readback Error Detection by Automatic Speech Recognition to Increase ATM Safety." In: *14th USA/Europe Air Traffic Management Research and Development Seminar, ATM 2021* (cit. on pp. 117, 118).

Helmke, Hartmut, Karel Ondřej, Shruthi Shetty, Hörður Arilíusson, Teodor S Simiganoschi, Matthias Kleinert, Oliver Ohneiser, Heiko Ehr, Juan-Pablo Zuluaga, and Pavel Smrz (2022). *Readback Error Detection by Automatic Speech Recognition and Understanding Results of HAAWAII project for Isavia's Enroute Airspace*. Tech. rep. URL: http://publications.idiap.ch/attachments/papers/2022/Helmke%7B%5C_%7DSIDS2022%7B%5C_%7D2022.pdf (cit. on pp. 12, 118, 126, 129).

Helmke, Hartmut, Michael Slotty, Michael Poiger, Damián Ferrer Herrer, Oliver Ohneiser, Nathan Vink, Aneta Cerna, Petri Hartikainen, Billy Josefsson, David Langr, Raquel García Lasheras, Gabriela Marin, Odd Georg Mevatne, Sylvain Moos, Mats N. Nilsson, and Mario Boyero Pérez (Dec. 2018). "Ontology for transcription of ATC speech commands of SESAR 2020 solution PJ.16-04." In: *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*. Vol. 2018-Septe. Institute of Electrical and Electronics Engineers Inc. ISBN: 9781538641125. DOI: 10.1109/DASC.2018.8569238 (cit. on p. 56).

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory." In: *Neural computation* 9.8, pp. 1735–1780 (cit. on p. 2).

Hofbauer, Konrad, Stefan Petrik, and Horst Hering (2008). "The ATCOSIM corpus of non-prompted clean air traffic control speech." In: *Proceedings of the 6th International Conference on Language Resources and Evaluation, LREC 2008*, pp. 2147–2152. ISBN: 2951740840. URL: http://www.lrec-conf.org/proceedings/lrec2008/pdf/545%7B%5C_%7Dpaper.pdf (cit. on pp. 56, 58, 101).

Holub, Alex, Pietro Perona, and Michael C Burl (2008). "Entropy-based active learning for object recognition." In: *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops*. ISBN: 9781424423408. DOI: 10.1109/CVPRW.2008.4563068 (cit. on p. 121).

Hopfield, J. J. (1982). "Neural networks and physical systems with emergent collective computational abilities." In: *Proceedings of the National Academy of Sciences of the United States of America* 79.8, pp. 2554–2558. ISSN: 00278424. DOI: 10.1073/pnas.79.8.2554 (cit. on p. 2).

Hsu, Wei Ning, Anuroop Sriram, Alexei Baevski, Tatiana Likhomanenko, Qiantong Xu, Vineel Pratap, Jacob Kahn, Ann Lee, Ronan Collobert, Gabriel Synnaeve, and Michael Auli (2021). "Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training." In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH* 3, pp. 2123–2127. ISSN: 19909772. DOI: 10.21437/Interspeech.2021-236. arXiv: 2104.01027 (cit. on pp. 6, 58).

Hu, Yuchen, Chen Chen, Qiushi Zhu, and Eng Siong Chng (2023). "Wav2code: Restore Clean Speech Representations via Codebook Lookup for Noise-Robust ASR." In: pp. 1–12. arXiv: 2304.04974. URL: http://arxiv.org/abs/2304.04974 (cit. on p. 58).

Huang, Lei, Jie Qin, Yi Zhou, Fan Zhu, Li Liu, and Ling Shao (2023). "Normalization Techniques in Training DNNs: Methodology, Analysis and Application." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.8, pp. 10173–10196. ISSN: 19393539. DOI: 10.1109/TPAMI.2023.3250241. arXiv: 2009.12836 (cit. on p. 26).

Huang, Zili, Marc Delcroix, Leibny Paola Garcia, Shinji Watanabe, Desh Raj, and Sanjeev Khudanpur (2022). "Computer Speech & Language Joint speaker diarization and speech recognition based on region." In: *Computer Speech & Language* 72.September 2021, p. 101316. ISSN: 0885-2308. DOI: 10.1016/j.csl.2021.101316. URL: https://doi.org/10.1016/j.csl.2021.101316 (cit. on pp. 75, 76).

Jan, Zohaib, Farhad Ahamed, Wolfgang Mayer, Niki Patel, Georg Grossmann, Markus Stumptner, and Ana Kuusk (2023). "Artificial intelligence for industry 4.0: Systematic review of applications, challenges, and opportunities." In: *Expert Systems with Applications* 216.June 2022, p. 119456. ISSN: 09574174. DOI: 10.1016/j.eswa.2022.119456. URL: https://doi.org/10.1016/j.eswa.2022.119456 (cit. on p. 2).

Ji, Ziwei, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung (2023). "Survey of Hallucination in Natural Language Generation." In: *ACM Computing Surveys* 55.12. ISSN: 15577341. DOI: 10.1145/3571730. arXiv: 2202.03629 (cit. on p. 101).

Ji, Ziwei, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung (2023). "Towards Mitigating Hallucination in Large Language Models via Self-Reflection." In: *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 1827–1843. ISBN: 9798891760615. arXiv: 2310.06271 (cit. on p. 136).

JIA, Guimin, Fangyuan CHENG, Jinfeng YANG, and Dan LI (Dec. 2018a). "Intelligent checking model of Chinese radiotelephony read-backs in civil aviation

air traffic control." In: *Chinese Journal of Aeronautics* 31.12, pp. 2280–2289. ISSN: 10009361. DOI: 10.1016/j.cja.2018.10.001 (cit. on p. 11).

– (Dec. 2018b). "Intelligent checking model of Chinese radiotelephony read-backs in civil aviation air traffic control." In: *Chinese Journal of Aeronautics* 31.12, pp. 2280–2289. ISSN: 10009361. DOI: 10.1016/j.cja.2018.10.001 (cit. on p. 118).

Johnson, Justin M. and Taghi M. Khoshgoftaar (2019). "Survey on deep learning with class imbalance." In: *Journal of Big Data* 6.1. ISSN: 21961115. DOI: 10.1186/s40537-019-0192-5. URL: https://doi.org/10.1186/s40537-019-0192-5 (cit. on p. 37).

Jordan, C and S D Brennen (1992). *Instantaneous self-assessment of workload technique (ISA)*. URL: https://skybrary.aero/sites/default/files/bookshelf/1963.pdf (visited on 07/11/2022) (cit. on p. 44).

Kai, Hiroto, Shinnosuke Takamichi, Sayaka Shiota, and Hitoshi Kiya (Jan. 2021). "Lightweight Voice Anonymization Based on Data-Driven Optimization of Cascaded Voice Modification Modules." In: *2021 IEEE Spoken Language Technology Workshop, SLT 2021 - Proceedings*, pp. 560–566. DOI: 10.1109/SLT48900.2021.9383535 (cit. on pp. 43, 45).

Kalyan, Katikapalli Subramanyam, Ajit Rajasekharan, and Sivanesan Sangeetha (2021). "AMMUS : A Survey of Transformer-based Pretrained Models in Natural Language Processing." In: pp. 1–42. arXiv: 2108.05542. URL: http://arxiv.org/abs/2108.05542 (cit. on pp. 26, 27).

Kanda, Naoyuki, Xiong Xiao, Yashesh Gaur, Xiaofei Wang, Zhong Meng, Zhuo Chen, and Takuya Yoshioka (2022). "Transcribe-To-Diarize: Neural Speaker Diarization for Unlimited Number of Speakers Using End-To-End Speaker-Attributed Asr." In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* 2022-May, pp. 8082–8086. ISSN:

15206149. DOI: 10.1109/ICASSP43922.2022.9746225. arXiv: 2110.03151 (cit. on p. 8).

Kasttet, Mohammed Saïd, Abdelouahid Lyhyaoui, Douae Zbakh, Adil Aramja, and Abderazzek Kachkari (2024). "Toward Effective Aircraft Call Sign Detection Using Fuzzy String-Matching between ASR and ADS-B Data." In: *Aerospace* 11.1. ISSN: 22264310. DOI: 10.3390/aerospace11010032 (cit. on p. 9).

Kim, Chanwoo and Richard M. Stern (2008). "Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis." In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pp. 2598–2601. ISSN: 19909772. DOI: 10.21437/interspeech.2008-644 (cit. on pp. 59, 77, 83, 151).

Kim, Jaehyeon, Jungil Kong, and Juhee Son (2021). "Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech." In: ISSN: 26403498. arXiv: 2106.06103. URL: http://arxiv.org/abs/2106.06103 (cit. on p. 59).

Kingma, Diederik P. and Jimmy Lei Ba (2015). "Adam: A method for stochastic optimization." In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15. arXiv: 1412.6980 (cit. on p. 22).

Kiranyaz, Serkan, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J. Inman (2021). "1D convolutional neural networks and applications: A survey." In: *Mechanical Systems and Signal Processing* 151, p. 107398. ISSN: 10961216. DOI: 10.1016/j.ymssp.2020.107398. arXiv: 1905.03554. URL: https://doi.org/10.1016/j.ymssp.2020.107398 (cit. on p. 24).

Kirwan, Barry, Alyson Evans, Laura Donohoe, Andy Kilner, Tab Lamoureux, Toby Atkinson, and Heather MacKendrick (1997). "Human Factors in the ATM System Design Life Cycle." In: *FAA/Eurocontrol ATM R&D*, pp. 1–21. URL: https://scholar.google.de/scholar?hl=en%7B%5C&%7Das%7B%5C_

%7Dsdt=0%7B%5C%%7D2C5%7B%5C&%7Dq=Human+Factors+in+the+ATM+System+Design+Life+Cycle%7B%5C&%7DbtnG= (cit. on p. 44).

Klakow, Dietrich and Jochen Peters (2002). "Testing the correlation of word error rate and perplexity." In: *Speech Communication* 38.1-2, pp. 19–28. ISSN: 01676393. DOI: 10.1016/S0167-6393(01)00041-3 (cit. on p. 22).

Kleinert, Matthias, Hartmut Helmke, Gerald Siol, Heiko Ehr, Aneta Cerna, Christian Kern, Dietrich Klakow, Petr Motlicek, Youssef Oualil, Mittul Singh, and Ajay Srinivasamurthy (Dec. 2018). "Semi-supervised adaptation of assistant based speech recognition models for different approach areas." In: *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*. Vol. 2018-Septe. Institute of Electrical and Electronics Engineers Inc. ISBN: 9781538641125. DOI: 10.1109/DASC.2018.8569879 (cit. on pp. 89, 90).

Kleinert, Matthias, Hartmut Helmke, Gerald Siol, Heiko Ehr, Michael Finke, Youssef Oualil, and Ajay Srinivasamurthy (2017). *Machine learning of controller command prediction models from recorded radar data and controller speech utterances*. Tech. rep. URL: https://www.malorca-project.de/wp/wp-content/uploads/SID%7B%5C_%7D2017%7B%5C_%7DMALORCA.pdf (cit. on p. 102).

Kocour, Martin, Karel Veselý, Alexander Blatt, Juan Zuluaga Gomez, Igor Szöke, and Jan Černocký (2021). "Boosting of contextual information in ASR for air-traffic call-sign recognition." In: pp. 2993–2997 (cit. on pp. 57, 60, 89, 91).

Kocour, Martin, 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 (Dec. 2022). "Automatic Processing Pipeline for Collecting and Annotating Air-Traffic Voice Communication Data." In: *Engineering Proceedings* 2.1, p. 8. DOI: 10.3390/engproc2021013008 (cit. on pp. 101, 102, 109, 117, 121).

Kornblith, Simon, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton (2019). "Similarity of neural network representations revisited." In: *36th International Conference on Machine Learning, ICML 2019* 2019-June, pp. 6156–6175. arXiv: 1905.00414 (cit. on p. 60).

Kožović, Dejan V., Dragan Đurđević, Mirko R. Dinulović, Saša Milić, and Boško P. Rašuo (2023). "Air Traffic Modernization and Control: ADS-B System Implementation Update 2022 – a Review." In: *FME Transactions* 51.1, pp. 117–130. ISSN: 2406128X. DOI: 10.5937/fme2301117K (cit. on p. 20).

Krishnan, Aravind, Jesujoba Alabi, and Dietrich Klakow (2023). "On the N-gram Approximation of Pre-trained Language Models." In: arXiv: 2306.06892. URL: http://arxiv.org/abs/2306.06892 (cit. on p. 56).

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks." In: *Advances in neural information processing systems* 25 (cit. on p. 2).

Landini, Federico, Ján Profant, Mireia Diez, and Lukáš Burget (2022). "Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: Theory, implementation and analysis on standard tasks." In: *Computer Speech and Language* 71, p. 101254. ISSN: 0885-2308. DOI: https://doi.org/10.1016/j.csl.2021.101254. URL: https://www.sciencedirect.com/science/article/pii/S0885230821000619 (cit. on p. 76).

Lecun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning." In: *Nature* 521.7553, pp. 436–444. ISSN: 14764687. DOI: 10.1038/nature14539 (cit. on p. 2).

Lehouillier, Thibault, Jérémy Omer, François Soumis, and Cyril Allignol (2014). "Interactions between Operations and Planning in Air Traffic Control." In: *ICRAT 2014, 6th International Conference on Research in Air Transportation* (cit. on p. 11).

Li, Jing, Aixin Sun, Jianglei Han, and Chenliang Li (2022). "A Survey on Deep Learning for Named Entity Recognition." In: *IEEE Transactions on Knowledge and Data Engineering* 34.1, pp. 50–70. ISSN: 15582191. DOI: 10.1109/TKDE.2020.2981314. arXiv: 1812.09449 (cit. on p. 30).

Li, Zewen, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou (2022). "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects." In: *IEEE Transactions on Neural Networks and Learning Systems* 33.12, pp. 6999–7019. ISSN: 21622388. DOI: 10.1109/TNNLS.2021.3084827. arXiv: 2004.02806 (cit. on p. 24).

Loewenthal, Kate Miriam, Michael Eysenck, Duncan Harris, Guy Lubitsh, Tessa Gorton, and Helen Bicknell (Apr. 2000). "Stress, distress and air traffic incidents: Job dysfunction and distress in airline pilots in relation to contextuallyassessed stress." In: *Stress Medicine* 16.3, pp. 179–183. ISSN: 07488386. DOI: 10.1002/(sici)1099-1700(200004)16:3<179::aid-smi851>3.0.co;2-4. URL: https://onlinelibrary.wiley.com/doi/10.1002/(SICI)1099-1700(200004)16:3%7B%5C%%7D3C179::AID-SMI851%7B%5C%%7D3E3.0.CO;2-4 (cit. on p. 41).

Luig, Johannes and Alois Sontacchi (2010). "Workload monitoring through speech analysis: Towards a system for air traffic control." In: *27th Congress of the International Council of the Aeronautical Sciences 2010, ICAS 2010*. Vol. 6, pp. 4729–4738. ISBN: 9781617820496 (cit. on pp. 5, 41, 42, 44).

Lynch, J and J Roberts (2021). "Aviation English Assessment and Training. Collegiate Aviation Review International." In: *Collegiate Aviation Review International* 39.2, pp. 26–42. URL: http://ojs.library.okstate.edu/osu/index.php/CARI/article/view/8216/7644 (cit. on p. 137).

Ma, Shuming, Dongdong Zhang, and Ming Zhou (July 2020). "A Simple and Effective Unified Encoder for Document-Level Machine Translation." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguis-

tics, pp. 3505–3511. DOI: 10.18653/v1/2020.acl-main.321. URL: https://www.aclweb.org/anthology/2020.acl-main.321 (cit. on p. 44).

Malik, Mishaim, Muhammad Kamran Malik, Khawar Mehmood, and Imran Makhdoom (2021). "Automatic speech recognition: a survey." In: *Multimedia Tools and Applications* 80.6, pp. 9411–9457. ISSN: 15737721. DOI: 10.1007/s11042-020-10073-7 (cit. on p. 32).

McCulloch, Warren S and Walter Pitts (1990). "A logical calculus of the ideas immanent in nervous activity (reprinted from bulletin of mathematical biophysics, vol 5, pg 115-133, 1943)." In: *Bulletin of Mathematical Biology* 52.1–2, pp. 99–115. ISSN: 0092-8240. URL: http://journals2.scholarsportal.info/pdf/00928240/v52i1-2/99%7B%5C_%7Dalcotiiina.xml (cit. on p. 1).

McKinzie, Brandon, Vaishaal Shankar, Joseph Cheng, Yinfei Yang, Jonathon Shlens, and Alexander Toshev (2023). "Robustness in Multimodal Learning under Train-Test Modality Mismatch." In: *Proceedings of Machine Learning Research* 202, pp. 24291–24303. ISSN: 26403498 (cit. on p. 137).

Mohan, Brij and Lal Srivastava (n.d.). *Speaker anonymization : representation, evaluation and formal guarantees*. Tech. rep. URL: https://tel.archives-ouvertes.fr/tel-03674540 (cit. on p. 5).

Monson, Brian B., Eric J. Hunter, Andrew J. Lotto, and Brad H. Story (2014). "The perceptual significance of high-frequency energy in the human voice." In: *Frontiers in Psychology* 5.JUN, pp. 1–11. ISSN: 16641078. DOI: 10.3389/fpsyg.2014.00587 (cit. on p. 32).

Morrow, Daniel, Alfred Lee, and Michelle Rodvold (1993). "Analysis of Problems in Routine Controller-Pilot Communication." In: *The International Journal of Aviation Psychology* 3.4, pp. 285–302. ISSN: 15327108. DOI: 10.1207/s15327108ijap0304_3 (cit. on pp. 11, 118).

Mumuni, Alhassan and Fuseini Mumuni (2022). "Data augmentation: A comprehensive survey of modern approaches." In: *Array* 16.August, p. 100258. ISSN:

25900056. DOI: 10.1016/j.array.2022.100258. URL: https://doi.org/10.1016/j.array.2022.100258 (cit. on p. 36).

Muñoz-de-Escalona, Enrique, Maria Chiara Leva, and José Juan Cañas (2024). "Mental Workload as a Predictor of ATCO's Performance: Lessons Learnt from ATM Task-Related Experiments." In: *Aerospace* 11.8. ISSN: 2226-4310. DOI: 10.3390/aerospace11080691. URL: https://www.mdpi.com/2226-4310/11/8/691 (cit. on pp. 3, 5).

Nadeau, David and Satoshi Sekine (2007). "A survey of named entity recognition and classification." In: *Lingvisticae Investigationes* 30.1, pp. 3–26 (cit. on p. 30).

Needleman, Saul B. and Christian D. Wunsch (1970). "A general method applicable to the search for similarities in the amino acid sequence of two proteins." In: *Journal of Molecular Biology* 48.3, pp. 443–453. ISSN: 0022-2836. DOI: https://doi.org/10.1016/0022-2836(70)90057-4. URL: https://www.sciencedirect.com/science/article/pii/0022283670900574 (cit. on p. 81).

Neuwirth, Rostam J. (2022). "The EU Artificial Intelligence Act." In: *The EU Artificial Intelligence Act* 0106. DOI: 10.4324/9781003319436 (cit. on p. 35).

Ngiam, Jiquan, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng (2011). "Multimodal deep learning." In: *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, pp. 689–696. arXiv: 2301.04856 (cit. on p. 29).

Nigmatulina, Iuliia, Rudolf Braun, Juan Zuluaga-Gomez, and Petr Motlicek (Aug. 2021). *Improving callsign recognition with air-surveillance data in air-traffic communication*. arXiv: 2108.12156. URL: http://arxiv.org/abs/2108.12156 (cit. on pp. 3, 9, 89, 101).

Nikšić, Lejla and Ebru Arıkan Öztürk (Apr. 2022). "U.S./Europe Comparison of Atc-Related Accidents and Incidents." In: *International Journal for Traffic*

and Transport Engineering 12.2, pp. 155–169. ISSN: 2217544X. DOI: 10.7708/ijtte2022.12(2).01 (cit. on pp. 3, 40, 116).

Nogueiras, Albino, Asunción Moreno, Antonio Bonafonte, and José B. Mariño (2001). "Speech emotion recognition using hidden Markov models." In: *EUROSPEECH 2001 - SCANDINAVIA - 7th European Conference on Speech Communication and Technology*, pp. 2679–2682. ISBN: 8790834100. URL: http://gps-tsc.upc.es/veu/ (cit. on p. 42).

Ohneiser, Oliver, Hartmut Helmke, Heiko Ehr, Hejar Gürlük, Michael Hössl, and Thorsten Mühlhausen (2021). "Air Traffic Controller Supportby Speech Recognition." In: *Advances in Human Aspects of Transportation: Part II* 16.July. DOI: 10.54941/ahfe100712 (cit. on pp. 3, 116).

Ohneiser, Oliver, Saeed Sarfjoo, Hartmut Helmke, Shruthi Shetty, Petr Motlicek, Matthias Kleinert, Heiko Ehr, and Šarunas Murauskas (2021). *Robust Command Recognition for Lithuanian Air Traffic Control Tower Utterances*. Tech. rep., pp. 266–270. DOI: 10.21437/Interspeech.2021-935 (cit. on p. 101).

OpenAI et al. (2023). "GPT-4 Technical Report." In: 4, pp. 1–100. arXiv: 2303.08774. URL: http://arxiv.org/abs/2303.08774 (cit. on pp. 2, 33).

Oualil, Youssef, Marc Schulder, Hartmut Helmke, Anna Schmidt, and Dietrich Klakow (2015). "Real-time integration of dynamic context information for improving automatic speech recognition." In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. Vol. 2015-Janua, pp. 2107–2111 (cit. on p. 89).

Owodunni, Abraham, Aditya Yadavalli, Chris Emezue, Tobi Olatunji, and Clinton Mbataku (2024). "AccentFold: A Journey through African Accents for Zero-Shot ASR Adaptation to Target Accents." In: *EACL 2024 - 18th Conference of the European Chapter of the Association for Computational Linguistics, Findings of EACL 2024*, pp. 2146–2161. arXiv: 2402.01152 (cit. on p. 138).

Panariello, Michele, Natalia Tomashenko, Xin Wang, Xiaoxiao Miao, Pierre Champion, Hubert Nourtel, Massimiliano Todisco, Nicholas Evans, Emmanuel Vincent, and Junichi Yamagishi (2024). "The VoicePrivacy 2022 Challenge: Progress and Perspectives in Voice Anonymisation." In: *IEEE/ACM Transactions on Audio Speech and Language Processing* 32, pp. 3477–3491. ISSN: 23299304. DOI: 10.1109/TASLP.2024.3430530 (cit. on pp. 5, 138).

Park, Tae Jin, Naoyuki Kanda, Dimitrios Dimitriadis, Kyu J. Han, Shinji Watanabe, and Shrikanth Narayanan (2022). "A review of speaker diarization: Recent advances with deep learning." In: *Computer Speech and Language* 72. ISSN: 10958363. DOI: 10.1016/j.csl.2021.101317. arXiv: 2101.09624 (cit. on pp. 8, 34, 75).

Pellegrini, Thomas, Jérôme Farinas, Estelle Delpech, and François Lancelot (Oct. 2019). "The airbus air traffic control speech recognition 2018 challenge: Towards ATC automatic transcription and call sign detection." In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. Vol. 2019-Septe, pp. 2993–2997. DOI: 10.21437/Interspeech.2019-1962. arXiv: 1810.12614. URL: http://arxiv.org/abs/1810.12614 (cit. on pp. 3, 9, 90, 101).

Perrault, Ray and Jack Clark (2024). *Introduction to the AI Index Report 2024* (cit. on p. 2).

Petrashchuk, Olena and Anna P Borowska (2019). "Comparison of Selected Aeronautical English Tests." In: *Language and Literary Studies of Warsaw* 9, pp. 217–238. ISSN: 23005726. URL: https://www.proquest.com/scholarly-journals/comparison-selected-aeronautical-english-tests/docview/2575544698 (cit. on p. 137).

Phang, Jason, Haokun Liu, and Samuel R. Bowman (2021). "Fine-Tuned Transformers Show Clusters of Similar Representations Across Layers." In: *BlackboxNLP 2021 - Proceedings of the 4th BlackboxNLP Workshop on Analyzing*

and Interpreting Neural Networks for NLP, pp. 529–538. DOI: 10.18653/v1/2021.blackboxnlp-1.42. arXiv: 2109.08406 (cit. on pp. 58, 69).

Politou, Eugenia, Efthimios Alepis, and Constantinos Patsakis (2018). "Forgetting personal data and revoking consent under the GDPR: Challenges and proposed solutions." In: *Journal of Cybersecurity* 4.1. ISSN: 20572093. DOI: 10.1093/cybsec/tyy001. URL: http://www.zdnet.com/article/google-well-track-your-offline-credit-card- (cit. on p. 41).

Povey, Daniel, Gilles Boulianne, Lukas Burget, Petr Motlicek, and Petr Schwarz (2011). "The kaldi speech recognition." In: *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding* January, pp. 1–4. URL: http://kaldi.sf.net/ (cit. on p. 89).

Prinzo, O Veronika, Alfred M Hendrix, and Ruby Hendrix (2009). *The Outcome of ATC Message Length and Complexity on En Route Pilot Readback Performance*. URL: http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA494551%5Cnpapers2://publication/uuid/CD305670-DB5A-472B-9F10-931B9845ADD2 (cit. on p. 117).

Raab, F.H., R. Caverly, R. Campbell, M. Eron, J.B. Hecht, A. Mediano, D.P. Myer, and J.L.B. Walker (2002). "HF, VHF, and UHF systems and technology." In: *IEEE Transactions on Microwave Theory and Techniques* 50.3, pp. 888–899. DOI: 10.1109/22.989972 (cit. on p. 17).

Radford, Alec, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever (2022). "Robust Speech Recognition via Large-Scale Weak Supervision." In: arXiv: 2212.04356. URL: https://github.com/openai/%20http://arxiv.org/abs/2212.04356 (cit. on pp. 2, 6, 33).

Ramshaw, Lance A and Mitchell P Marcus (1999). "Text chunking using transformation-based learning." In: *Natural language processing using very large corpora*. Springer, pp. 157–176 (cit. on p. 31).

Ray, Partha Pratim (2023). "ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope." In: *Internet of Things and Cyber-Physical Systems* 3.March, pp. 121–154. ISSN: 26673452. DOI: 10.1016/j.iotcps.2023.04.003. URL: https://doi.org/10.1016/j.iotcps.2023.04.003 (cit. on p. 2).

Reimers, Nils and Iryna Gurevych (Aug. 2019). "Sentence-BERT: Sentence embeddings using siamese BERT-networks." In: *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*. Association for Computational Linguistics, pp. 3982–3992. ISBN: 9781950737901. DOI: 10.18653/v1/d19-1410. arXiv: 1908.10084. URL: http://arxiv.org/abs/1908.10084 (cit. on pp. 102, 104).

Rekkas, Christos (2014). "Status of WAM, ADS-B out and ATSAW deployment in Europe." In: *2014 Tyrrhenian International Workshop on Digital Communications - Enhanced Surveillance of Aircraft and Vehicles, TIWDC/ESAV 2014*, pp. 1–5. DOI: 10.1109/TIWDC-ESAV.2014.6945438 (cit. on p. 20).

Ren, Pengzhen, Yun Xiao, Xiaojun Chang, Po Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang (2022). "A Survey of Deep Active Learning." In: *ACM Computing Surveys* 54.9. ISSN: 15577341. DOI: 10.1145/3472291. arXiv: 2009.00236 (cit. on p. 37).

Request, User, Evaluation Tool, Traffic Management Advisory, Terminal Approach, Radar Control, Air Route, and Traffic Control (2006). "Sehchang Hah , Ph . D ., Ben Willems , M . A . & Randy Phillips , Supervisory Air Traffic Control Specialist * Human Factors Group-Atlantic City Federal Aviation Administration (FAA) Atlantic City International Airport , New Jersey." In: *Control*, pp. 50–54 (cit. on p. 3).

Rothe, Sascha, Shashi Narayan, and Aliaksei Severyn (2020). *Leveraging pre-trained checkpoints for sequence generation tasks*. Tech. rep., pp. 264–280. DOI:

10.1162/tacl_a_00313. arXiv: 1907.12461. URL: https://github.com/openai/gpt-2. (cit. on p. 94).

Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams (2019). "Learning Representations by Back-Propagating Errors." In: *Cognitive Modeling* 2, pp. 213–222. DOI: 10.7551/mitpress/1888.003.0013 (cit. on p. 2).

Sammito, Stefan, B. Thielmann, R. Seibt, A. Klussmann, M. Weippert, and I. Böckelmarn (2016). "Nutzung der herzschlagfrequenz und der herzfrequenzvariabilität in der arbeitsmedizin und der arbeitswissenschaft." In: *Arbeitsmedizin Sozialmedizin Umweltmedizin* 51.2, pp. 123–141. ISSN: 23634669 (cit. on p. 41).

Sarkar, Amitrajit, Surajit Dasgupta, Sudip Kumar Naskar, and Sivaji Bandyopadhyay (2018). "Says who? Deep learning models for joint speech recognition, segmentation and diarization." In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. Vol. 2018-April. IEEE, pp. 5229–5233. ISBN: 9781538646588. DOI: 10.1109/ICASSP.2018.8462375 (cit. on p. 74).

Schäfer, Matthias, Martin Strohmeier, Vincent Lenders, Ivan Martinovic, and Matthias Wilhelm (2014). "Bringing up OpenSky: A large-scale ADS-B sensor network for research." In: *IPSN 2014 - Proceedings of the 13th International Symposium on Information Processing in Sensor Networks (Part of CPS Week)*. IEEE Computer Society, pp. 83–94. ISBN: 9781479931460. DOI: 10.1109/IPSN.2014.6846743 (cit. on pp. 20, 91).

Schmidt, Anna, Youssef Oualil, Oliver Ohneiser, Matthias Kleinert, Marc Schulder, Arif Khan, Hartmut Helmke, and Dietrich Klakow (2014). "Context-based recognition network adaptation for improving on-line ASR in air traffic control." In: *2014 IEEE Workshop on Spoken Language Technology, SLT 2014 - Proceedings*, pp. 13–15. ISBN: 9781479971299. DOI: 10.1109/SLT.2014.7078542 (cit. on p. 89).

Sebastian, Glorin (2023). "Privacy and Data Protection in ChatGPT and Other AI Chatbots." In: *International Journal of Security and Privacy in Pervasive Computing* 15.1, pp. 1–14. ISSN: 2643-7937. DOI: 10.4018/ijsppc.325475 (cit. on p. 35).

Serafini, Luca, Samuele Cornell, Giovanni Morrone, Enrico Zovato, Alessio Brutti, and Stefano Squartini (2023). "An experimental review of speaker diarization methods with application to two-speaker conversational telephone speech recordings." In: *Computer Speech and Language* 82. ISSN: 10958363. DOI: 10.1016/j.csl.2023.101534. arXiv: 2305.18074 (cit. on p. 34).

Shafey, Laurent El, Hagen Soltau, and Izhak Shafran (2019). "Joint speech recognition and speaker diarization via sequence transduction." In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH* 2019-Septe, pp. 396–400. ISSN: 19909772. DOI: 10.21437/Interspeech.2019-1943. arXiv: 1907.05337 (cit. on pp. 8, 74, 75, 80).

Sharma, Ashwani, Tarun Virmani, Vipluv Pathak, Anjali Sharma, Kamla Pathak, Girish Kumar, and Devender Pathak (2022). "Artificial Intelligence-Based Data-Driven Strategy to Accelerate Research, Development, and Clinical Trials of COVID Vaccine." In: *BioMed Research International* 2022. ISSN: 23146141. DOI: 10.1155/2022/7205241 (cit. on p. 2).

Shin, Hyeon Kyeong, Hyewon Han, Kyungguen Byun, and Hong Goo Kang (2020). "Speaker-invariant Psychological Stress Detection Using Attention-based Network." In: *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2020 - Proceedings*, pp. 308–313. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=%7B%5C&%7Darnumber=9306384 (cit. on pp. 42, 44, 46, 47).

Shore, Todd, Friedrich Faubel, Hartmut Helmke, and Dietrich Klakow (2012a). "Knowledge-based word lattice rescoring in a dynamic context." In: *13th Annual Conference of the International Speech Communication Association 2012, IN-*

TERSPEECH 2012. Vol. 2, pp. 1082–1085. ISBN: 9781622767595. DOI: 10.21437/interspeech.2012-328. URL: http://www.isca-speech.org/archive (cit. on p. 3).

Shore, Todd, Friedrich Faubel, Hartmut Helmke, and Dietrich Klakow (2012b). "Knowledge-based word lattice rescoring in a dynamic context." In: *13th Annual Conference of the International Speech Communication Association 2012, INTERSPEECH 2012*. Vol. 2, pp. 1082–1085. ISBN: 9781622767595 (cit. on p. 89).

Sillard, L., F. Vergne, and B. Desart (2000). "LES DIFFERENTS TYPES DE STRESS PROFESSIONNEL APPLIQUES AU CONTROLE AERIEN." In: *European Organisation for the Safety of Air Navigation, EUROCONTROL* 12.13, pp. 93–129. ISSN: 0014-0139. URL: http://www.eurocontrol.int/eec/public/standard%7B%5C_%7Dpage/DOC%7B%5C_%7DReport%7B%5C_%7D2006%7B%5C_%7D023.html (cit. on p. 41).

Silver, David, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George Van Den Driessche, Thore Graepel, and Demis Hassabis (2017). "Mastering the game of Go without human knowledge." In: *Nature* 550.7676, pp. 354–359. ISSN: 14764687. DOI: 10.1038/nature24270. URL: http://dx.doi.org/10.1038/nature24270 (cit. on p. 2).

Sleeman, William C, Rishabh Kapoor, and Preetam Ghosh (2021). "Multimodal Classification: Current Landscape, Taxonomy and Future Directions." In: arXiv: 2109.09020. URL: http://arxiv.org/abs/2109.09020 (cit. on p. 29).

Song, Hwanjun, Minseok Kim, Dongmin Park, Yooju Shin, and Jae Gil Lee (2023). "Learning From Noisy Labels With Deep Neural Networks: A Survey." In: *IEEE Transactions on Neural Networks and Learning Systems* 34.11, pp. 8135–8153. ISSN: 21622388. DOI: 10.1109/TNNLS.2022.3152527. arXiv: 2007.08199 (cit. on p. 36).

Sousa, Samuel and Roman Kern (2023). *How to keep text private? A systematic review of deep learning methods for privacy-preserving natural language processing*. Vol. 56. 2. Springer Netherlands, pp. 1427–1492. ISBN: 1046202210204. DOI: 10.1007/s10462-022-10204-6. arXiv: 2205.10095. URL: https://doi.org/10.1007/s10462-022-10204-6 (cit. on p. 138).

Srinivasamurthy, A., P. Motlicek, M. Singh, Y. Oualil, M. Kleinert, H. Ehr, and H. Helmke (2018). "Iterative learning of speech recognition models for air traffic control." In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH* 2018-Septe.September, pp. 3519–3523. ISSN: 19909772. DOI: 10.21437/Interspeech.2018-1447 (cit. on pp. 89, 90, 100).

Srivastava, Brij Mohan Lal, Mohamed Maouche, Md Sahidullah, Emmanuel Vincent, Aurelien Bellet, Marc Tommasi, Natalia Tomashenko, Xin Wang, and Junichi Yamagishi (2022). "Privacy and Utility of X-Vector Based Speaker Anonymization." In: *IEEE/ACM Transactions on Audio Speech and Language Processing* 30, pp. 2383–2395. ISSN: 23299304. DOI: 10.1109/TASLP.2022.3190741. URL: https://hal.inria.fr/hal-03197376v3 (cit. on p. 35).

Stevens, S. S., J. Volkmann, and E. B. Newman (1937). "A Scale for the Measurement of the Psychological Magnitude Pitch." In: *Journal of the Acoustical Society of America* 8.3, pp. 185–190. ISSN: NA. DOI: 10.1121/1.1915893 (cit. on p. 46).

Tager-Flusberg, Helen (2015). "The Development of English as a Second Language With and Without Specific Language Impairment: Clinical Implications." In: *Journal of Speech, Language, and Hearing Research* 24.2, pp. 1–14. ISSN: 1096-6218. DOI: 10.1044/2015. arXiv: 0803973233 (cit. on p. 65).

Tiewtrakul, T and SR Fletcher (2010). "The challenge of regional accents for aviation English language proficiency standards: A study of difficulties in understanding in air traffic control–pilot communications." In: *Ergonomics* 53.2, pp. 229–239 (cit. on p. 17).

Tomashenko, Natalia, Brij Mohan Lal Srivastava, Xin Wang, Emmanuel Vincent, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Jose Patino, Jean-François Bonastre, Paul-Gauthier Noé, and Massimiliano Todisco (2022). "The VoicePrivacy 2020 Challenge Evaluation Plan." In: 6.Table 1. arXiv: 2205.07123. URL: http://arxiv.org/abs/2205.07123 (cit. on p. 36).

Tomashenko, Natalia, Xin Wang, Emmanuel Vincent, Jose Patino, Brij Mohan Lal Srivastava, Paul Gauthier Noé, Andreas Nautsch, Nicholas Evans, Junichi Yamagishi, Benjamin O'Brien, Anaïs Chanclu, Jean François Bonastre, Massimiliano Todisco, and Mohamed Maouche (2022). *The VoicePrivacy 2020 Challenge: Results and findings*. Tech. rep. DOI: 10.1016/j.csl.2022.101362. arXiv: 2109.00648. URL: https://www.voiceprivacychallenge.org/ (cit. on pp. 42, 43).

Tomba, Kevin, Joel Dumoulin, Elena Mugellini, Omar Abou Khaled, and Salah Hawila (2018). "Stress detection through speech analysis." In: *ICETE 2018 - Proceedings of the 15th International Joint Conference on e-Business and Telecommunications* 1, pp. 394–398. DOI: 10.5220/0006855803940398 (cit. on p. 42).

Uclés, Natalia Rodríguez and José Manuel Cordero García (2014). "Relationship between Workload and Duration of ATC Voice Communications." In: *Icrat*, pp. 1–8 (cit. on p. 5).

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017a). "Attention is all you need." In: *Advances in Neural Information Processing Systems*. Vol. 2017-Decem. arXiv, pp. 5999–6009. DOI: 10.48550/ARXIV.1706.03762. arXiv: 1706.03762. URL: https://arxiv.org/abs/1706.03762 (cit. on pp. 2, 25, 26, 118, 145).

– (June 2017b). "Attention is all you need." In: *Advances in Neural Information Processing Systems*. Vol. 2017-Decem. Neural information processing systems foundation, pp. 5999–6009. arXiv: 1706.03762 (cit. on p. 23).

Vergin, Rivarol, Douglas O'Shaughnessy, and Azarshid Farhat (1999). "Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition." In: *IEEE Transactions on Speech and Audio Processing* 7.5, pp. 525–532. ISSN: 10636676. DOI: 10.1109/89.784104 (cit. on p. 32).

Wang, Qi, Yue Ma, Kun Zhao, and Yingjie Tian (2022). "A Comprehensive Survey of Loss Functions in Machine Learning." In: *Annals of Data Science* 9.2, pp. 187–212. ISSN: 21985812. DOI: 10.1007/s40745-020-00253-5. URL: https://doi.org/10.1007/s40745-020-00253-5 (cit. on p. 21).

Wolf, Thomas et al. (2020). "Transformers: State-of-the-Art Natural Language Processing." In: pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6. arXiv: 1910.03771v5. URL: https://github.com/huggingface/ (cit. on p. 93).

Wu, Qiong, Brett R.C. Molesworth, and Dominique Estival (Apr. 2019). "An Investigation into the Factors that Affect Miscommunication between Pilots and Air Traffic Controllers in Commercial Aviation." In: *International Journal of Aerospace Psychology* 29.1-2, pp. 53–63. ISSN: 24721832. DOI: 10.1080/24721840.2019.1604138 (cit. on p. 118).

Xia, Wei, Han Lu, Quan Wang, Anshuman Tripathi, Yiling Huang, Ignacio Lopez Moreno, and Hasim Sak (2022). "Turn-To-Diarize: Online Speaker Diarization Constrained By Transformer Transducer Speaker Turn Detection." In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* 2022-May.2, pp. 8077–8081. ISSN: 15206149. DOI: 10.1109/ICASSP43922.2022.9746531. arXiv: 2109.11641 (cit. on pp. 8, 75, 76).

Xu, Mingke, Fan Zhang, Xiaodong Cui, and Wei Zhang (2021). "Speech Emotion Recognition with Multiscale Area Attention and Data Augmentation." In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* 2021-June, pp. 6319–6323. ISSN: 15206149. DOI: 10.1109/ICASSP39728.2021.9414635. arXiv: 2102.01813 (cit. on p. 42).

Yang, Hui Hua, Yu Hern Chang, and Yi Hui Chou (2023). "Subjective measures of communication errors between pilots and air traffic controllers." In: *Journal of Air Transport Management* 112.July, p. 102461. ISSN: 09696997. DOI: 10.1016/j.jairtraman.2023.102461. URL: https://doi.org/10.1016/j.jairtraman.2023.102461 (cit. on p. 11).

Yoo, In Chul, Keonnyeong Lee, Seonggyun Leem, Hyunwoo Oh, Bonggu Ko, and Dongsuk Yook (2020). "Speaker Anonymization for Personal Information Protection Using Voice Conversion Techniques." In: *IEEE Access* 8, pp. 198637–198645. ISSN: 21693536. DOI: 10.1109/ACCESS.2020.3035416 (cit. on p. 35).

Zafar, Afia, Muhammad Aamir, Nazri Mohd Nawi, Ali Arshad, Saman Riaz, Abdulrahman Alruban, Ashit Kumar Dutta, and Sultan Almotairi (2022). "A Comparison of Pooling Methods for Convolutional Neural Networks." In: *Applied Sciences (Switzerland)* 12.17, pp. 1–21. ISSN: 20763417. DOI: 10.3390/app12178643 (cit. on p. 25).

Zeng, Weili, Xiao Chu, Zhengfeng Xu, Yan Liu, and Zhibin Quan (2022). "Aircraft 4D Trajectory Prediction in Civil Aviation: A Review." In: *Aerospace* 9.2. ISSN: 22264310. DOI: 10.3390/aerospace9020091 (cit. on p. 137).

Zhang, Wei, Maryam Kamgarpour, Dengfeng Sun, and Claire J. Tomlin (2012). "A hierarchical flight planning framework for air traffic management." In: *Proceedings of the IEEE* 100.1, pp. 179–194. ISSN: 00189219. DOI: 10.1109/JPROC.2011.2161243 (cit. on p. 3).

Zheng, F., G. Zhang, and Z. Song (2001). "Comparison of different implementations of MFCC." In: *Journal of Computer Science and Technology* 16.6, pp. 582–589. ISSN: 10009000. DOI: 10.1007/BF02943243 (cit. on p. 32).

Zhu, Dawei, Michael A Hedderich, Fangzhou Zhai, David Ifeoluwa Adelani, and Dietrich Klakow (2022). "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*, pp. 62–

67. ISBN: 9781955917407. DOI: 10.18653/v1/2022.insights-1.8. arXiv: 2204.09371 (cit. on pp. 37, 119, 126).

Zhu, Qiu Shi, Jie Zhang, Zi Qiang Zhang, Ming Hui Wu, Xin Fang, and Li Rong Dai (2022). "a Noise-Robust Self-Supervised Pre-Training Model Based Speech Representation Learning for Automatic Speech Recognition." In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* 2022-May.62101523, pp. 3174–3178. ISSN: 15206149. DOI: 10.1109/ICASSP43922.2022.9747379. arXiv: 2201.08930 (cit. on p. 58).

Zollinger, Sue Anne and Henrik Brumm (2011). "The Lombard effect." In: *Current Biology* 21.16, R614–R615. ISSN: 09609822. DOI: 10.1016/j.cub.2011.06.003. URL: http://dx.doi.org/10.1016/j.cub.2011.06.003 (cit. on p. 43).

Zuluaga-Gomez, Juan, Petr Motlicek, Qingran Zhan, Karel Vesely, and Rudolf Braun (June 2020). "Automatic speech recognition benchmark for air-traffic communications." In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. Vol. 2020-Octob, pp. 2297–2301. DOI: 10.21437/Interspeech.2020-2173. arXiv: 2006.10304. URL: http://dx.doi.org/10.21437/Interspeech.2020-2173 (cit. on pp. 56, 57, 60, 89, 92).

Zuluaga-Gomez, Juan, Amrutha Prasad, Iuliia Nigmatulina, Saeed Sarfjoo, Petr Motlicek, Matthias Kleinert, Hartmut Helmke, Oliver Ohneiser, and Qingran Zhan (Mar. 2022). "How Does Pre-trained Wav2Vec2.0 Perform on Domain Shifted ASR? An Extensive Benchmark on Air Traffic Control Communications." In: arXiv: 2203.16822. URL: http://arxiv.org/abs/2203.16822 (cit. on p. 6).

Zuluaga-Gomez, Juan, Amrutha Prasad, Iuliia Nigmatulina, Seyyed Saeed Sarfjoo, Petr Motlicek, Matthias Kleinert, Hartmut Helmke, Oliver Ohneiser, and Qingran Zhan (2023a). "How Does Pre-Trained Wav2Vec 2.0 Perform on Domain-Shifted Asr? an Extensive Benchmark on Air Traffic Control Communications." In: *2022 IEEE Spoken Language Technology Workshop, SLT 2022*

- *Proceedings*, pp. 205–212. DOI: 10.1109/SLT54892.2023.10022724. arXiv: 2203.16822 (cit. on p. 3).

Zuluaga-Gomez, Juan, Amrutha Prasad, Iuliia Nigmatulina, Seyyed Saeed Sarfjoo, Petr Motlicek, Matthias Kleinert, Hartmut Helmke, Oliver Ohneiser, and Qingran Zhan (2023b). "How Does Pre-Trained Wav2Vec 2.0 Perform on Domain-Shifted Asr? an Extensive Benchmark on Air Traffic Control Communications." In: *2022 IEEE Spoken Language Technology Workshop, SLT 2022 - Proceedings*, pp. 205–212. DOI: 10.1109/SLT54892.2023.10022724. arXiv: 2203.16822 (cit. on pp. 56, 57, 60).

Zuluaga-Gomez, Juan, Seyyed Saeed Sarfjoo, Amrutha Prasad, Iuliia Nigmatulina, Petr Motlicek, Karel Ondrej, Oliver Ohneiser, and Hartmut Helmke (2023). "Bertraffic: Bert-Based Joint Speaker Role and Speaker Change Detection for Air Traffic Control Communications." In: *2022 IEEE Spoken Language Technology Workshop, SLT 2022 - Proceedings*, pp. 633–640. DOI: 10.1109/SLT54892.2023.10022718. arXiv: 2110.05781 (cit. on pp. 3, 7, 34, 76, 79).

Zuluaga-Gomez, Juan, Karel Veselý, Alexander Blatt, Petr Motlicek, Dietrich Klakow, Allan Tart, Igor Szöke, Amrutha Prasad, Saeed Sarfjoo, Pavel Kolčárek, Martin Kocour, Honza Černocký, Claudia Cevenini, Khalid Choukri, Mickael Rigault, and Fabian Landis (Dec. 2020a). "Automatic Call Sign Detection: Matching Air Surveillance Data with Air Traffic Spoken Communications." In: *Proceedings* 59.1, p. 14. DOI: 10.3390/proceedings2020059014 (cit. on pp. 59, 76, 77).

– (2020b). "Automatic Call Sign Detection: Matching Air Surveillance Data with Air Traffic Spoken Communications." In: *Proceedings* 59.1. ISSN: 2504-3900. DOI: 10.3390/proceedings2020059014. URL: https://www.mdpi.com/2504-3900/59/1/14 (cit. on p. 100).

Zuluaga-Gomez, Juan, Karel Veselý, Igor Szöke, Alexander Blatt, Petr Motlicek, Martin Kocour, Mickael Rigault, Khalid Choukri, Amrutha Prasad, Seyyed Saeed Sarfjoo, Iuliia Nigmatulina, Claudia Cevenini, Pavel Kolčárek, Allan Tart,

Jan Černocký, and Dietrich Klakow (2022). "ATCO2 corpus: A Large-Scale Dataset for Research on Automatic Speech Recognition and Natural Language Understanding of Air Traffic Control Communications." In: pp. 1–29. arXiv: 2211.04054. URL: http://arxiv.org/abs/2211.04054 (cit. on pp. 19, 56, 58, 76, 117).

Declaration

I hereby declare that this dissertation is my own original work except where otherwise indicated. All data or concepts drawn directly or indirectly from other sources have been correctly acknowledged. This dissertation has not been submitted in its present or similar form to any other academic institution either in Germany or abroad for the award of any other degree.

Saarbrücken, April 15, 2025

Alexander Blatt