Low-Resource Speech Recognition and Understanding for Challenging Applications

Thèse n. 10616
Présenté le 14 Juin 2024
à la Faculté des sciences et techniques de l'ingénieur
Laboratoire de l'IDIAP
Programme doctoral en Génie Électrique et Électronique
École Polytechnique Fédérale de Lausanne

pour l'obtention du grade de Docteur dès Sciences
par

Juan Pablo ZULUAGA-GOMEZ

acceptée sur proposition du jury:

Prof. Andrei POPESCU-BELIS, président du jury
Prof. Jean-Philippe THIRAN, directeur de thèse
Dr. Petr MOTLICEK, co-directeur de thèse
Dr. Philip GARNER, rapporteur interne
Prof. Yannick ESTÈVE, rapporteur externe
Prof. Jan ČERNOCKÝ, rapporteur externe

Lausanne, EPFL, August 28, 2024

Acknowledgements

My PhD journey has been a remarkable experience, filled with joyful moments that made my time in Martigny and Lausanne truly special–not just academically, but socially as well. These four years have flown by, and I am deeply grateful to the many people who have been an integral part of this chapter in my life.

First, I express my gratitude to my supervisor, **Petr Motlicek**, for his mentorship, support, advice and help throughout these four years. His constructive and helpful feedback helped me to largely improve my professional and academic skills. I could see the tiny, albeit constant and incremental, improvements across the years. Also, I am deeply thankful for his support while exploring multiple research directions.

I express my thanks to the jury members of my thesis, Prof. **Andrei POPESCU-BELIS**, Dr. **Philip GARNER**, Prof. **Yannick ESTÈVE**, Prof. **Jan ČERNOCKÝ**, and Prof. **Jean-Philippe THIRAN**, for dedicating their time to review this thesis and for their constructive feedback. I am also grateful to the Idiap secretariats and administrative staff for facilitating my stay in Switzerland and for fostering a conducive research environment.

I am also thankful to **Ernie Pusateri** from Apple Boston and **David**, **Xing**, **Sundar** and **Marcello** from Amazon-AWS Seattle, whose guidance and collaboration during my two internships in US profoundly enriched my knowledge on speech recognition and translation and familiarized me with the intricacies of the industry environment.

I am thankful too to my colleagues–which became friends–from Idiap, where I shared incredible moments, such as hiking, trekking, camping, skiing, SUPing, biking, traveling, Caves Ouverting, picnicking, etc. Across the years, I shared many moments with a first batch of friends, specially with **Julian**, **Apoorv** & **Sargam**, **François**, **Florian Piras** (my Parce and an invaluable friend), **Florian Mai**, **Weipeng**, **Suraj**, **Suhan**, **Neha**, **Laurent**, **Amir**, **Zohreh**, **Eklavya**, **Tilak**, **Bogdan**, **Chloe**, **Louise**, **Enno**, **Yulia**, **Colombine**, **Esau** and **Sergio**. Then, and due to the changing nature of a research institute, I had the chance to become friend of incredible people, including **Fabio** (aka Faf), **Andrei**, **Anshul**, **Arya** & **Pierre**, **Darya**, **Alina**, **Olena**, **Karl**, **Shashi** & **Yacouba**, **Roberto**, **Mirko**, **Maxime**, **Laura**, etc. Special thanks to **Barbara** for the last six months of nice talks, walks, and train rides! Some names might be missing, but all of them are in my heart. All these friends, that influenced who I am today, and that took a key role in shaping my career, thank you all! Special thanks to my ex-flatmates **Neha** and **Suhan**. I feel lucky to have shared time at Idiap and at home, *it truly felt like home*.

I also feel incredibly fortunate to have met **Alex** (my Parce!) and later **Valentina**. They were a constant source of support during the challenging times of my PhD, always engaging in conversations about science as well as lighter topics. From discussing AI, servers, microchip manufacturing, and OpenAI to debating Venezuelan versus Colombian arepas and pan de bonos, their companionship made a significant difference in these years. I'll always cherish the wonderful memories of cooking Reina Pepiada together after running multiple times in Lausanne.

Also, I am thankful to my "Latin" friends in Lausanne, which supported me throughout my PhD. Specially, thanks to **Rafa** (**Noah** & **Simona**), **Franco**, **Alfredo**, **Andres**, **Josie**, and **Tahel**. Also, to many more "Latin/Colombian" friends across Europe, such as **Sergio**, **Marco**, **Nico**, **Hernán** and **David**, I feel truly lucky to have met them.

Reflecting on my PhD journey, I express my gratitude to my first mentor in research, **Pablo Bonaveri** (Universidad Autonoma del Caribe). He was the first person who sparked my interest in research. Without his advice, support and time, I would not have taken the path of a Master's Degree, which later lead to a PhD. I am thankful too to all my previous professors and teachers who partly had a role in shaping who I am today.

Finally, I extend my heartfelt thanks to my family back in Colombia. **Nora**, **Omar**, **Daniela**, **Esteban**, and **Paula**. For their unwavering support, advice, encouragement, and mostly, love, throughout this challenging yet rewarding journey. Without them, this would not have been possible. I love you all.

"Switzerland is the best country to pursue a PhD"

Lausanne, August 28, 2024 Juan Pablo Zuluaga Gomez

Abstract

Automatic speech recognition (ASR) and spoken language understanding (SLU) is the core component of current voice-powered AI assistants such as Siri and Alexa. It involves speech transcription with ASR and its comprehension with natural language understanding (NLU) systems. Traditionally, SLU runs on a cascaded setting, where an in-domain ASR system automatically generates the transcripts with valuable semantic information, e.g., named entities and intents. These components have been generally based on statistical approaches with hand-crafted features. However, current trends have shifted towards large-scale end-to-end (E2E) deep neural networks (DNN), which have shown superior performance on a wide range of SLU tasks. For example, ASR has seen a rapid transition from traditional hybrid-based modeling to encoder-decoder and Transducer-based modeling. Even though there is an undeniable improvement in performance, other challenges have come into play, such as the urgency and need of large-scale supervised datasets; the need of additional modalities, such as contextual knowledge; massive GPU clusters for training large models; or high-performance and robust large models for complex applications. All of this leads to major challenges. This thesis explores solutions to these challenges that arise from complex settings. Specifically, we propose approaches: (1) to overcome the data scarcity on hybrid-based and E2E ASR models, i.e., low-resource applications; (2) for integration of contextual knowledge at decoding and training time, which leads to improved model quality; (3) to fast develop streaming ASR models from scratch for challenging domains without supervised data; (4) to reduce the computational budget required at training and inference time by proposing efficient alternatives w.r.t the state-of-the-art E2E architectures. Similarly, we explore solutions on the SLU domain, including analysis on the optimal representations to perform cascaded SLU, and other SLU tasks aside from intent and slot filing that can be performed in an E2E fashion. Finally, this thesis closes by covering `STAC-ST` and `TokenVerse`, two novel architectures that can handle ASR and SLU tasks seamlessly in a single model via special tokens.

Keywords: Automatic Speech Recognition, Spoken Language Understanding, Conversational Speech, Air Traffic Control Communications, End-to-End ASR, Low-Resource ASR.

Résumé

La reconnaissance automatique de la parole (ASR) et la compréhension du langage parlé (SLU) sont au cœur des assistants d'intelligence artificielle à commande vocale actuels, tels que Siri et Alexa. Elle implique la transcription de la parole avec la ASR et sa compréhension avec des systèmes de NLU. Traditionnellement, le SLU fonctionne en cascade, où un système ASR dans le domaine génère automatiquement les transcriptions avec des informations sémantiques précieuses, par exemple les entités nommées et les intentions. Ces composants ont généralement été basés sur des approches statistiques avec des caractéristiques créées à la main. Cependant, les tendances actuelles se sont orientées vers les DNN à grande échelle de E2E, qui ont montré des performances supérieures sur un large éventail de tâches SLU. Par exemple, l'ASR a connu une transition rapide de la modélisation traditionnelle basée sur les hybrides à la modélisation basée sur les encoder-decoder et les Transducers. Même si l'amélioration des performances est indéniable, d'autres défis sont entrés en jeu, comme l'urgence de données supervisées à grande échelle, le besoin de modalités supplémentaires, telles que la connaissance contextuelle, les clusters GPU massifs pour l'entraînement de grands modèles, ou les grands modèles performants et robustes pour les applications complexes. Tout ceci conduit à des défis majeurs. Cette thèse explore les solutions à ces défis qui découlent de contextes complexes. Plus précisément, nous proposons des approches : (1) pour surmonter la rareté de données sur les modèles ASR hybrides et E2E, les applications à faibles ressources ; (2) pour l'intégration de la connaissance contextuelle au moment du décodage et de l'entraînement, ce qui permet d'améliorer la qualité du modèle ; (3) pour développer rapidement des modèles ASR en continu à partir de zéro pour les domaines difficiles sans données supervisées ; (4) pour réduire le budget de GPU requis au moment de l'entraînement et de l'inférence en proposant des alternatives efficaces par rapport aux architectures E2E les plus récentes. De même, nous explorons des solutions dans le domaine du SLU, y compris l'analyse des représentations optimales pour effectuer le SLU en cascade, et d'autres tâches SLU en dehors de l'intention et du classement des créneaux qui peuvent être effectuées d'une manière E2E. Enfin, cette thèse se termine par l'étude de `STAC-ST` et `TokenVerse`, deux nouvelles architectures qui peuvent traiter les tâches ASR et SLU de manière transparente dans un modèle unique via des jetons spéciaux.

Mots-clés : ASR, SLU, Langage conversationnel, communications de contrôle du trafic aérien (ATC), End-to-End ASR, ASR à ressources limitées.

Contents

Acronyms

AC	Aho-Corasick
AI	Artificial Intelligence
AM	Acoustic Model
AED	Attention-Based Encoder Decoder
ATC	Air Traffic Control
ATM	Air Traffic Management
ASR	Automatic Speech Recognition
ANSPs	Air Navigation Service Providers
ATCo	Air Traffic Controller
ATCC	Air Traffic Control Communication
ADS-B	Automatic Dependent Surveillance–Broadcast
BPE	Byte-Pair Encoding
CER	Character Error Rate
CTC	Connectionist Temporal Classification
CNN	Convolutional Neural Network
Conformer	Convolution-Augmented Transformer
dB	Decibel
DER	Diarization Error Rate
DNN	Deep Neural Networks
E2E	End-To-End
ELD	English Language Detection
ENDP	Endpointing
ELDA	European Language Resources Association
FST	Finite State Transducer
FSM	Foundational Speech Model
ICAO	International Civil Aviation Organization
GMM	Gaussian Mixture Model
GELU	Gaussian Error Linear Units
HMM	Hidden Markov model
JER	Jaccard Error Rate
KD	Knowledge Distillation

LF-MMI	Lattice-Free Maximum Mutual Information
LM	Language Model
ML	Machine Learning
MT	Machine Translation
MLP	Multilayer Perceptron
MFFCs	Mel-frequency Cepstral Coefficients
MHSA	Multi-Head Self-Attention
NE	Named Entity
NER	Named Entity Recognition
NLL	Negative Log-Likelihood
NLP	Natural Language Processing
NLU	Natural Language Understanding
OOD	Out-of-Domain
OOV	Out-of-Vocabulary
OSN	OpenSky Network
PL	Pseudo-Label
PER	Phoneme Error Rate
PTT	Push-To-Talk
RTX	Real-Time Factor
RNN	Recurrent Neural Network
RNN-T	RNN-Transducer
SC	Sequence Classification
SD	Speaker Diarization
SF	Shallow Fusion
ST	Speech-to-Text Translation
SCD	Speaker Change Detection
SLU	Spoken Language Understanding
SSL	Self-Supervised Learning
SST	Self-Supervised Training
SNR	Signal-To-Noise
SOT	Serialized Output Training
SRD	Speaker Role Detection
TT	Transformer-Transducer
TDNN	Time Delay Neural Network
TDNNF	Factorized TDNN
VAD	Voice Activity Detection
VHF	Very-High Frequency
WCN	Word Confusion Network
WER	Word Error Rate
WFST	Weighted Finite State Transducer

List of Figures

List of Tables

1 Introduction

In the evolving landscape of speech technology, foundational speech models (FSMs) have become the key component to unify various speech tasks such as automatic speech recognition (ASR) and spoken language understanding (SLU). Multilingual settings have also shown promising performance, particularly in well-defined benchmarks and databases. However, despite this progress, FSMs encounter challenges in certain scenarios. For instance, their robustness in low-latency, low-resourced and complicated applications, such as the ones with limited supervised data and compute, remains a concern.

Questions arise about the feasibility of developing ASR systems without supervised data and the effective extraction of relevant information from spoken conversations using SLU techniques. Additionally, the impracticality of FSMs on streaming settings motivates for further exploration. This thesis addresses these questions from multiple perspectives. In the following section, we summarize motivations and challenges of current ASR and SLU systems.

1.1 Motivation

Not enough supervised data The necessity to achieve high accuracy and low WERs with limited supervised data indicates the need for innovative approaches in multiple speech tasks. Fine-tuning from large pretrained FSMs emerge as viable solutions to mitigate the scarcity of supervised data. Additionally, augmenting ASR systems with additional modalities during training and decoding holds the potential for enhanced performance. In this thesis, we explore the integration of contextual knowledge (e.g., surveillance or radar data in the domain of air traffic control dialogues) as an extra modality interfaced with ASR systems during training and decoding to improve overall WERs.

Usage of contextual information In pursuit of reducing WERs on low-resource settings, leveraging contextual information becomes essential. Traditional methods for improving WERs often entail costly architecture modifications or large supervised training datasets. However, in this thesis, we propose to leverage contextual data to enhance certain n-grams at ASR inference time, thus aiding the recognition of rare words without incurring on substantial architectural changes or increased training data.

Fast Streaming ASR prototyping Real-life industrial applications demand rapid development of ASR systems, often with limited in-domain supervised data and with the need to run on low-latency streaming fashion. This poses significant challenges, including (1) in-domain data scarcity, (2) the inherent complexity of streaming ASR w.r.t offline decoding, and (3) strict time constraints for model development. To address these challenges, we propose several approaches: (1) reducing the time and supervised data required for ASR development via knowledge distillation, (2) proposing a linearized alternative of the attention mechanism in Transformers to improve training efficiency and decoding speed, (3) employing high quality pseudo-labeled data from FSMs to overcome data scarcity, and (4) introducing the attention sink mechanisms within the ASR field to improve performance in challenging low-latency streaming scenarios without compromising decoding speed.

Spoken language understanding In many industrial applications, ASR serves as a preliminary–intermediary–step before performing higher-level NLU or SLU tasks. We address multiple NLU/SLU tasks, including intent detection, slot filling, and speaker role detection, both in cascaded and end-to-end formats, with and without reliance on intermediate ASR hypotheses.

Enabling multitasking with special tokens End-to-end joint ASR and SLU models offer compelling advantages, such as reduced parameter counts and unified optimization process across multiple downstream tasks. However, challenges arise concerning paired data availability for various tasks. Our contributions include demonstrating the feasibility of training attention-based encoder-decoder and transducer models for multiple task by leveraging special tokens, enabling decoding of ASR, speech-to-text translation, and acoustic named-entity recognition within a unified framework.

1.2 Thesis Outline

The outline of this thesis is summarized below by chapter.

Chapter 2 In this chapter, we provide a gentle introduction to automatic speech recognition and its two most prominent paradigms: hybrid-based ASR and end-to-end ASR. Next, we review spoken language understanding and the two methods currently used, the cascaded and the end-to-end pipeline. Later, we examine the three domains and applications targeted in this thesis, including (1) read and prompted speech; (2) conversational speech; and (3) air traffic control communications. Finally, we provide a comprehensive overview of the evaluation metrics utilized throughout this thesis, to evaluate ASR and SLU systems across various tasks and domains.

Chapter 3 In this chapter, the focus lies on challenging ASR applications constrained by the availability of supervised data, particularly in air traffic control (ATC) communications. Benchmarking ASR for ATC with open-source databases is introduced, revealing the existing gap between large-scale ASR systems and niche applications like ATC. We also propose strategies for leveraging pretrained FSMs to overcome data scarcity, along with innovative approaches to incorporate contextual information (e.g., surveillance or user data) during decoding. Furthermore, we propose an approach to leverage contextual information for improved semi-supervised training on ATC speech under low resource settings.

Chapter 4 Afterward, we tackle training-and-compute-bounded challenges in ASR, particularly for conversational speech. Novel methods for rapidly developing transducer-based streaming ASR solutions are presented, leveraging FSMs through sequence-level knowledge distillation. Effective techniques for data selection and filtering are introduced to mitigate errors propagated from pseudo-labels, enhancing training efficiency and reducing computation time while achieving lower WERs. Additionally, an adaptation of semi-supervised learning-based models to the transducer architecture, termed XLSR-Transducer, is proposed. We close the chapter with the introduction of HyperConformer, a novel architecture that achieves comparable or superior ASR recognition performance compared to Conformer while exhibiting greater efficiency in terms of inference speed, memory usage, parameter count, and availability of training data.

Chapter 5 Here, we explore advancements in SLU for challenging applications such as ATC communications. We explore various downstream tasks such as slot filling, callsign highlighting, and joint speaker role and change detection. Then we perform a comprehensive benchmarking of text, acoustic, and lattice-based representations for intent and slot-filling on a challenging database for in-home personal robot assistants (SLURP).

Chapter 6 Finally, the thesis concludes by examining joint ASR and SLU architectures where we optimize a single model for multiple tasks via task tokens that condition the models at training and decoding time. Specifically, we propose solutions for two prominent E2E architectures: (1) attention-based encoder-decoder and (2) transducer-based architectures. This chapter cover the following tasks: multilingual ASR and speech-to-text translation, cross-talk detection, and acoustic-based speaker turn detection. All of these tasks are of large relevance, especially for industrial applications, that might require low-latency solutions.

1.3 Publications

This thesis is a compilation of 3 journal publications and 14 conference publications where I am first author or contributed significantly:

Journal papers (published or submitted):

1. J. Zuluaga-Gomez, A. Prasad, I. Nigmatulina, P. Motlicek, and M. Kleinert, "A Virtual Simulation-Pilot Agent for Training of Air Traffic Controllers," *Aerospace*, vol. 10, no. 5, p. 490, 2023
2. J. Zuluaga-Gomez, I. Nigmatulina, A. Prasad, P. Motlicek, D. Khalil, S. Madikeri, A. Tart, I. Szoke, V. Lenders, M. Rigault, *et al.*, "Lessons Learned in Transcribing 5000 h of Air Traffic Control Communications for Robust Automatic Speech Understanding," *Aerospace*, vol. 10, no. 10, p. 898, 2023
3. J. Zuluaga-Gomez, K. Veselý, I. Szöke, A. Blatt, P. Motlicek, M. Kocour, M. Rigault, K. Choukri, A. Prasad, S. S. Sarfjoo, *et al.*, "ATCO2 Corpus: A Large-Scale Dataset for Research on Automatic Speech Recognition and Natural Language Understanding of Air Traffic Control Communications," *Submitted to Data-centric Machine Learning Research (DMLR) Journal, arXiv preprint arXiv:2211.04054*, 2024

Conference papers (published, submitted, or to be submitted):

1. J. Zuluaga-Gomez, S. Kumar, *et al.*, "Improved Streaming Transformer Transducer With Attention Sinks," in *To be Submitted to ARR (long paper)*, 2024
2. I. Nigmatulina, J. Zuluaga-Gomez, *et al.*, "Fast Streaming Transducer ASR Prototyping via Knowledge Distillation with Whisper," in *Submitted to EMNLP 2024 (long paper).* ***[Equal contribution]***, 2024
3. J. Zuluaga-Gomez, Z. Huang, X. Niu, R. Paturi, S. Srinivasan, P. Mathur, B. Thompson, and M. Federico, "End-to-End Single-Channel Speaker-Turn Aware Conversational Speech Translation," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 7255–7274
4. I. Nigmatulina, J. Zuluaga-Gomez, *et al.*, "Improved contextual adaptation with an external n-gram language model for Transducer-based ASR," in *Submitted to INTERSPEECH 2024*, 2024

5. S. Kumar, S. Madikeri, J. Zuluaga-Gomez, I. Nigmatulina, E. Villatoro-Tello, S. Burdisso, P. Motlicek, K. Pandia, and A. Ganapathiraju, “TokenVerse: Unifying Speech and NLP Tasks via Transducer-based ASR,” in *arXiv:2407.04444*, 2024
6. S. Kumar, S. Madikeri, J. Zuluaga-Gomez, E. Villatoro-Tello, I. Nigmatulina, P. Motlicek, M. K. E, and A. Ganapathiraju, “XLSR-Transducer: Streaming ASR for Self-Supervised Pretrained Models,” in *arXiv:2407.04439*, 2024
7. F. Mai, J. Zuluaga-Gomez, T. Parcollet, and P. Motlicek, “HyperConformer: Multi-head HyperMixer for Efficient Speech Recognition,” in *Proc. Interspeech*, 2023, pp. 2213–2217
8. J. Zuluaga-Gomez, A. Prasad, I. Nigmatulina, S. S. Sarfjoo, P. Motlicek, M. Kleinert, H. Helmke, O. Ohneiser, and Q. Zhan, “How Does Pre-trained Wav2Vec 2.0 Perform on Domain-Shifted ASR? An Extensive Benchmark on Air Traffic Control Communications,” in *IEEE Spoken Language Technology Workshop (SLT).* IEEE, 2023, pp. 205–212
9. J. Zuluaga-Gomez, S. S. Sarfjoo, A. Prasad, I. Nigmatulina, P. Motlicek, K. Ondrej, O. Ohneiser, and H. Helmke, “BERTRAFFIC: Bert-based joint speaker role and speaker change detection for air traffic control communications,” in *IEEE Spoken Language Technology Workshop (SLT).* IEEE, 2023, pp. 633–640
10. I. Nigmatulina, J. Zuluaga-Gomez, A. Prasad, S. S. Sarfjoo, and P. Motlicek, “A two-step approach to leverage contextual data: speech recognition in air-traffic communications,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2022, pp. 6282–6286
11. A. Prasad, J. Zuluaga-Gomez, P. Motlicek, S. Sarfjoo, I. Nigmatulina, and K. Veselý, “Speech and Natural Language Processing Technologies for Pseudo-Pilot Simulator,” in *12th SESAR Innovation Days.* Sesar Joint Undertaking., 2022
12. J. Zuluaga-Gomez, I. Nigmatulina, A. Prasad, P. Motlicek, K. Veselý, M. Kocour, and I. Szöke, “Contextual Semi-Supervised Learning: An Approach to Leverage Air-Surveillance and Untranscribed ATC Data in ASR Systems,” in *Proc. Interspeech*, 2021, pp. 3296–3300
13. J. Zuluaga-Gomez, K. Veselý, A. Blatt, P. Motlicek, D. Klakow, A. Tart, I. Szöke, A. Prasad, S. Sarfjoo, P. Kolčárek, *et al.*, “Automatic call sign detection: Matching air surveillance data with air traffic spoken communications,” in *Proceedings*, vol. 59, no. 1. MDPI, 2020, p. 14
14. J. Zuluaga-Gomez, P. Motlicek, Q. Zhan, K. Veselý, and R. Braun, “Automatic Speech Recognition Benchmark for Air-Traffic Communications,” in *Proc. Interspeech*, 2020, pp. 2297–2301

In addition to the papers above, the following 8 papers, to which I contributed and that are relevant to this thesis:

1. J. Zuluaga-Gomez, S. Ahmed, D. Visockas, and C. Subakan, “CommonAccent: Exploring Large Acoustic Pretrained Models for Accent Classification Based on Common Voice,” in *Proc. Interspeech*, 2023, pp. 5291–5295
2. M. Kocour, K. Veselý, A. Blatt, J. Zuluaga-Gomez, I. Szöke, J. Černocký, D. Klakow, and P. Motlicek, “Boosting of Contextual Information in ASR for Air-Traffic Call-Sign

Recognition," in *Proc. Interspeech*, 2021, pp. 3301–3305

3. M. Kocour, K. Veselý, I. Szöke, S. Kesiraju, J. Zuluaga-Gomez, A. Blatt, A. Prasad, I. Nigmatulina, P. Motlíček, D. Klakow, *et al.*, "Automatic processing pipeline for collecting and annotating air-traffic voice communication data," *Engineering Proceedings*, vol. 13, no. 1, p. 8, 2021
4. M. Rigault, C. Cevenini, K. Choukri, M. Kocour, K. Veselý, I. Szoke, P. Motlicek, J. Zuluaga-Gomez, A. Blatt, D. Klakow, *et al.*, "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*, 2022, pp. 79–83
5. H. Helmke, K. Ondřej, S. Shetty, H. Arilíusson, T. S. Simiganosch, M. Kleinert, O. Ohneiser, H. Ehr, and J. Zuluaga-Gomez, "Readback Error Detection by Automatic Speech Recognition and Understanding-Results of HAAWAII project for Isavia's Enroute Airspace," *12th SESAR Innovation Days.*, 2022
6. H. Helmke, M. Kleinert, N. Ahrenhold, H. Ehr, T. Mühlhausen, O. Ohneiser, L. Klamert, P. Motlicek, A. Prasad, J. Zuluaga-Gomez, *et al.*, "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)*, 2023
7. I. Nigmatulina, S. Madikeri, E. Villatoro-Tello, P. Motlicek, J. Zuluaga-Gomez, K. Pandia, and A. Ganapathiraju, "Implementing Contextual Biasing in GPU Decoder for Online ASR," in *Proc. Interspeech*, 2023, pp. 4494–4498
8. E. Villatoro-Tello, S. Madikeri, J. Zuluaga-Gomez, B. Sharma, S. S. Sarfjoo, I. Nigmatulina, P. Motlicek, A. V. Ivanov, and A. Ganapathiraju, "Effectiveness of text, acoustic, and lattice-based representations in spoken language understanding tasks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5

In addition to the papers above, I contributed to 9 additional papers that are either published on workshops, journal or pre-print servers (not included in the thesis for space reasons):

1. M. Ravanelli, T. Parcollet, A. Moumen, S. de Langen, C. Subakan, P. Plantinga, Y. Wang, P. Mousavi, L. D. Libera, A. Ploujnikov, F. Paissan, D. Borra, S. Zaiem, Z. Zhao, S. Zhang, G. Karakasidis, S.-L. Yeh, A. Rouhe, R. Braun, F. Mai, J. Zuluaga-Gomez, *et al.*, "Open-Source Conversational AI with SpeechBrain 1.0," in *arXiv:2407.00463*, 2024
2. I. Nigmatulina, R. Braun, J. Zuluaga-Gomez, and P. Motlicek, "Improving callsign recognition with air-surveillance data in air-traffic communication," Idiap Research Institute. Idiap Research Institute, 2021, pp. 1–5
3. D. Khalil, A. Prasad, P. Motlicek, J. Zuluaga-Gomez, I. Nigmatulina, S. Madikeri, and C. Schuepbach, "An Automatic Speaker Clustering Pipeline for the Air Traffic Communication Domain," *Aerospace*, vol. 10, no. 10, p. 876, 2023

4. N. Ahrenhold, H. Helmke, T. Mühlhausen, O. Ohneiser, M. Kleinert, H. Ehr, L. Klamert, and J. Zuluaga-Gómez, "Validating Automatic Speech Recognition and Understanding for Pre-Filling Radar Labels—Increasing Safety While Reducing Air Traffic Controllers' Workload," *Aerospace*, vol. 10, no. 6, p. 538, 2023
5. S. Burdisso, J. Zuluaga-Gomez, E. Villatoro-Tello, M. Fajcik, M. Singh, P. Smrz, and P. Motlicek, "IDIAPers@ Causal News Corpus 2022: Efficient Causal Relation Identification Through a Prompt-based Few-shot Approach," in *Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE)*, 2022, pp. 61–69
6. M. Fajcik, M. Singh, J. Zuluaga-Gomez, E. Villatoro-Tello, S. Burdisso, P. Motlicek, and P. Smrz, "IDIAPers@ Causal News Corpus 2022: Extracting Cause-Effect-Signal Triplets via Pre-trained Autoregressive Language Model," in *Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE)*, 2022, pp. 70–78
7. A. Prasad, J. Zuluaga-Gomez, P. Motlicek, S. Sarfjoo, I. Nigmatulina, O. Ohneiser, and H. Helmke, "Grammar Based Speaker Role Identification for Air Traffic Control Speech Recognition," *12th SESAR Innovation Days.*, 2022
8. Q. Zhan, X. Xie, C. Hu, J. Zuluaga-Gomez, J. Wang, and H. Cheng, "Domain-Adversarial Based Model with Phonological Knowledge for Cross-Lingual Speech Recognition," *Electronics*, vol. 10, no. 24, p. 3172, 2021
9. S. Madikeri, S. Tong, J. Zuluaga-Gomez, A. Vyas, P. Motlicek, and H. Bourlard, "Pkwrap: a pytorch package for lf-mmi training of acoustic models," *arXiv preprint arXiv:2010.03466*, 2020

1.4 Contributions to Projects

This thesis contains a set of contributions that can be categorized into multiple research and innovation projects.

1.4.1 ATCO2

In ATCO2 project, I participated on data collection and curation. ASR training for both, hybrid-based and E2E models. Also, I proposed multiple NLU systems for ATC, such as callsign and command extraction, speaker role detection and text-based speaker diarization. Finally, we released multiple datasets under the ATCO2 project for different ASR and NLU tasks.

The work was supported by European Union's Horizon 2020 project No. 864702 - ATCO2 (Automatic collection and processing of voice data from air-traffic communications), which is a part of Clean Sky Joint Undertaking.

1.4.2 HAAWAII

In HAAWAII project, my main contribution was on ASR training for both, hybrid-based and E2E models.

The work was supported by European Union's Horizon 2020 project No. 884287 - HAAWAII (Highly automated air-traffic controller workstations with artificial intelligence integration).

1.4.3 EUROCONTROL

In the industrial EUROCONTROL project, I contributed to the development of a pseudo-pilot system [16], which can aid the training of ATCos by integrating ASR and NLU tools in the learning process. The first version of the pseudo-pilot system was presented in Paris, France in 2022. This work was enlarged with a publication in the Aerospace journal [3].

1.4.4 Uniphore

In the industrial Uniphore project, I worked on large-scale pseudo-labeling of speech with foundational speech models, focused on conversational speech for call-center use cases. Additionally, I contributed with ASR streaming solutions for call-center speech in English. Section 4.1, Section 4.2 and Section 4.3 are my contributions to this project.

2 Background

In this chapter, we cover foundational background on automatic speech recognition (ASR) and spoken language understanding (SLU), the primary domains of interest in this thesis. We start with a comprehensive overview of the fundamental components and paradigms of ASR and SLU, including both cascaded and end-to-end methodologies. After, we define the challenging applications that serve as the domains covered in this thesis. Lastly, we discuss the primary evaluation metrics employed to assess the performance of these systems.

2.1 Automatic Speech Recognition

Automatic speech recognition (ASR) is an interdisciplinary research field that aims to develop techniques and methods that allow the recognition and translation of spoken language into text. Generally, recorded speech is represented as a sequence of acoustic feature vectors or *observations*: $\boldsymbol{X}$, whereas the output word sequence is represented by $\boldsymbol{W}$. During recognition or *decoding*, the main goal is finding the most likely $\boldsymbol{W}$ given the input sequence $\boldsymbol{X}$. Traditionally, this task is addressed with statistical models trained on a labeled *corpus with audio-text pairs*, as $D = \{\boldsymbol{X^n}, \boldsymbol{W^n}\}$. The most likely word sequence ($\hat{\boldsymbol{W}}$) can be modeled as:

$$\hat{\boldsymbol{W}} = arg \max_{\boldsymbol{W}} \boldsymbol{P}(\boldsymbol{W}|\boldsymbol{X}). \tag{2.1}$$

The problem is further expanded using Bayes Theorem:

$$\boldsymbol{P}(\boldsymbol{W}|\boldsymbol{X}) = \frac{P_{AM}(\boldsymbol{X}|\boldsymbol{W})P_{LM}(\boldsymbol{W})}{P(\boldsymbol{X})}, \tag{2.2}$$

where, $P_{AM}(\boldsymbol{X}|\boldsymbol{W})$ stands for the likelihood of the feature sequence $\boldsymbol{X}$, given the word sequence $\boldsymbol{W}$. This term is normally denoted as the acoustic model (AM). The language model (LM), $P_{LM}(\boldsymbol{W})$, denotes the probability of the word sequence. $P(\boldsymbol{X})$ is the a-priori probability

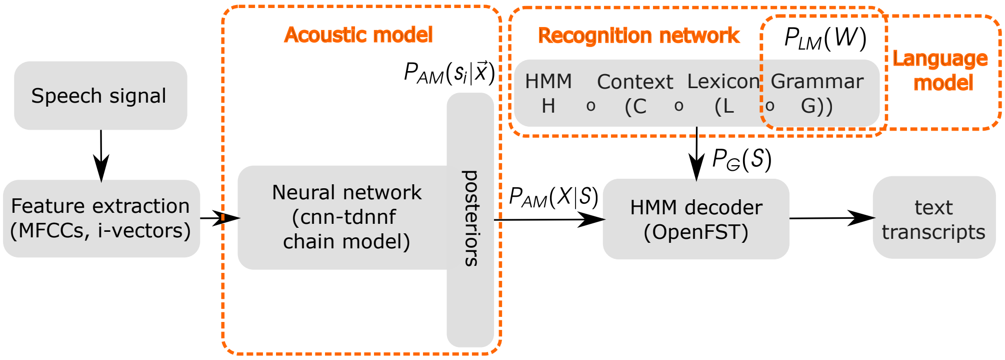


Figure 2.1: Hybrid-based ASR system architecture. The inference pipeline consists of *feature extraction*, *acoustic matching* by acoustic model and *search* by HMM decoder that uses HCLG recognition network. On the output are text transcripts. The output can be a *lattice* or *confusion network*. From [5].

of the feature sequence $\boldsymbol{X}$, but it is ignored during the maximization operation due to its independence from the word sequence. Equation 2.3 is further simplified as $P(\boldsymbol{X})$ is a constant for any word sequence, as follows:

$$\hat{\boldsymbol{W}} = arg \max_{\boldsymbol{W}} P_{AM}(\boldsymbol{X}|\boldsymbol{W})\ P_{LM}(\boldsymbol{W}). \tag{2.3}$$

To summarize, the ASR systems rely on an acoustic and language model as stated above. Currently, there are two main ASR paradigms, where different strategies, architectures, and procedures are employed for blending all these modules in one “system" [37, 38].

2.1.1 Hybrid-based Modeling

Automatic speech recognition with hybrid systems is based on hidden Markov models (HMM) and deep neural networks (DNN). DNNs are an effective module for the estimating the posterior probability of a given set of possible outputs (e.g., phone- or tri-phone state). These posterior probabilities can be seen as pseudo-likelihoods or “scale likelihoods”, which can be interfaced with HMM modules [39, 40]. HMMs provide a structure for mapping a temporal sequence of acoustic features, e.g., Mel-frequency cepstral coefficients (MFCCs) or mel-filter banks (Fbanks) into a sequence of states [41, 40].

Hybrid systems have a separated pronunciation lexicon, language model, and acoustic model, as shown in Figure 2.1. They are optimized separately, which allows more freedom to estimate the right set of hyper-parameters of each module. This includes the integration of more resources to build better LM or lexicons.

Lexicon *The lexicon* is a table that maps words into pronunciations (phoneme-strings). It is a resource used by the HMM-based ASR systems. Numerous ASR engines used the CMU Pronouncing Dictionary (lexicon),[1] which defines the phoneme set, and it is used as the training data for the grapheme-to-phoneme (G2P) module that synthesizes pronunciations of "new words". Example of a pronunciation for a 'spelled acronyms' like "KLM" is represented as "k ey eh l eh m". *The vocabulary* is the set of finite possible words that an ASR systems can generate, while each word has a pronunciation mapped from the lexicon. In general, the lexicon can be viewed as a Finite State Transducer (FST). One key advantage of hybrid systems versus other ASR paradigms is that the text data (e.g., words, dictionary) and pronunciation of new words are added before training, hoping to match the target domain.

Inference with hybrid-based ASR To generate a hypothesized transcript from the input sequence of features, the scores of AM and LM are combined using a decoding graph. We aim at finding the most likely word sequence $\hat{\boldsymbol{W}}$ (transcription), in the matrix of acoustic scores. The search explores HMM paths that exist in a recognition network, termed *HCLG graph* (see Figure 2.1). The standard decoding algorithm is based on two ideas: *token passing* [42] and *beam search* [43]. The search combines scores from the AM, LM and lexicon, where the Equation 2.3 gets into:

$$\hat{\boldsymbol{W}} = arg \max_{\boldsymbol{S}} P_{AM}(S|\boldsymbol{X})^{\kappa} \; P_G(S) \tag{2.4}$$

In Equation 2.4, the AM scores are the model posteriors $P_{AM}(S|\boldsymbol{X})$, where $\boldsymbol{X}$ is the time-series of input features and S is an HMM state-sequence. The language model and lexicon scores are both represented in the graph score $P_G(S)$ that is present in the HCLG recognition network. κ is an empirical scaling constant, for chain models the optimal $\kappa = 1.0$.

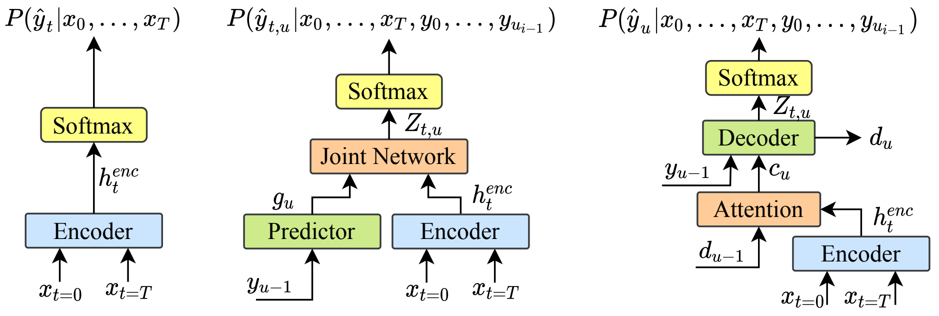
The HCLG graph The HCLG graph is a Weighted Finite State Transducer (WFST). It is composed of a language model graph G, pronunciation lexicon graph L, context dependency graph C and phoneme HMM graphs H. The HCLG graph contains graph costs P_G that originate from its source graphs, while the most important source is the language model [44].

Overall, hybrid systems still remain one of the best and most flexible approaches for building ASR engines when relatively low amount of audio-text pairs is available for training. In contrast, with the large growth of computational power and unlabeled data, new ASR paradigms have emerged. This includes E2E ASR, which do not rely on HMMs and do not require complicated pronunciation lexicons.

[1] Dictionary at: http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

2.1.2 End-to-End Modeling

End-to-end (E2E) speech recognition aims at directly transcribing speech to text without requiring alignments between acoustic frames (i.e., input features) and output characters/words. In hybrid systems, this is a key step where lattices from a previous Gaussian Mixture Model (GMM) and HMM model and alignments created on the fly are required to train the ASR system. Unlike the hybrid approaches, the E2E model learns a direct mapping between acoustic frames and text units (e.g., subword units or characters) or words in a single step towards the final objective of interest. Finally, E2E systems attempt to bypass the suboptimal issues that arise from training separately the AM and LM. In Figure 2.2 we list the three more prominent E2E architecture.



(a) CTC **(b) Transducer** **(c) Encoder-decoder**

Figure 2.2: Top three prominent end-to-end ASR architectures. (a) CTC; (b) Transducer, and (c) attention-based encoder–decoder.

CTC-based modeling

Connectionist Temporal Classification (CTC) [45, 46] is a sequence discriminative training criterion used in ASR. Unlike methods requiring frame-level alignments between input feature sequences $\boldsymbol{X}_u$ and target label sequences y_u, CTC circumvents this need. Analogous to the state sequence definition in HMMs, CTC is initiated by establishing the notion of a path between X_u and y_u. A plausible path is constructed by extending the label sequence y_u of size U, to match the input acoustic length of T. This extension involves the repetition of any label and/or the insertion of the blank symbol (ϕ),[2] i.e., no label. Collectively, this results in what is termed the CTC path. There have been numerous work on CTC-based ASR [47, 48, 49, 50, 51], whereas the model architecture is in Figure 2.2 (a). Despite the simplicity of CTC models, there are two problems: (1) The output sequence length U has to be smaller than the input sequence length T; (2) output at timestamp t is assumed to be independent, e.g., of x_{t-1}, and x_{t+1} [52].

[2] Blank symbol denotes emitting no label at a given time step t.

Neural Transducer based ASR

Neural Transducer, usually termed Recurrent Neural Network-Transducer (RNN-T)[3] is a sequence-to-sequence model [53]. Transducer models solve both problems of CTC (§2.1.2). First, it allows multiple label outputs at each timestamp, and second, it adds a predictor network that acts as a weak language model, given context based on the previous decoding steps. In a typical neural transducer model (Figure 2.2b) there are three networks: the encoder, predictor, and joiner [54]. The encoder processes audio frames to produce acoustic embeddings. The predictor generates token embeddings in an auto-regressive manner, taking previous non-blank tokens as input. Lastly, the joiner combines the outputs from the encoder and predictor to predict a probability distribution over the tokens in the vocabulary. Overall, the output can be written as:

$$\mathbf{X}_{0:t} = \text{Encoder}(\mathbf{x}_{0:t}), \tag{2.5}$$

$$\mathbf{Y}_{1:u} = \text{Predictor}(\mathbf{y}_{1:u}), \tag{2.6}$$

$$\text{Joint} = \text{Linear}(\mathbf{X}_{0:t}) + \text{Linear}(\mathbf{Y}_{1:u}), \tag{2.7}$$

$$P(y_{u+1}|\mathbf{X}_{0:t}, \mathbf{Y}_{1:u}) = \text{Softmax}(\text{Joiner}(\text{Joint})), \tag{2.8}$$

where $\mathbf{X}_{0:t}$ is the output from encoder; $\mathbf{Y}_{1:u}$ are the token-embeddings from Predictor; and P is the probability of predicting y_{u+1} given past tokens and audio embeddings as input. Recent work aiming at reducing the computational requirements of transducer models, utilize a stateless predictor [55] network, i.e., no RNNs or Transformer layers are required. It is composed of an embedding layer and one 1-D Convolution Neural Network (CNN) layer. Finally, the joiner network consists of one linear layer. Furthermore, the encoders, such as LSTM [56], Conformer [57], Zipformer [58] or FastConformer [59] are trained from scratch and require large amount of in-domain supervised data to achieve acceptable WER. The Transducer models employ the vanilla RNN-T loss [53], but more efficient variants such as pruned-transducer loss are also used, which is a memory-efficient alternative to standard RNN-T loss [60].[4]

Within the transducer framework, the usage of Transformer [61] encoders leads to the Transformer-Transducer (TT) [62, 63]. This architecture is a popular choice for streaming ASR [64, 65, 66] because of its robustness and low WERs.

Encoder-Decoder modeling

Attention-based encoder-decoder (AED) models are a growing family of models that first ‘encode’ the input features into a higher dimensional space of an “embedding” by the encoder block (h_t^{enc}). Then, the decoder block classifies the “latent features” (C_u) into a sequence of tokens defined by the vocabulary.[5] AED models are optimized to learn the audio-text alignment directly. An

[3] Even though it is widely known as RNN-T, it does not require RNNs. For instance, Transformer-based encoders are frequently used.

[4] Available at k2 toolkit: https://github.com/k2-fsa/k2.

[5] In most recent work, word-based vocabularies [8, 10] have been replaced by sub-word units, e.g., byte-pair encoding (BPE) [67] or sentence piece [68] or even character-level vocabularies [49].

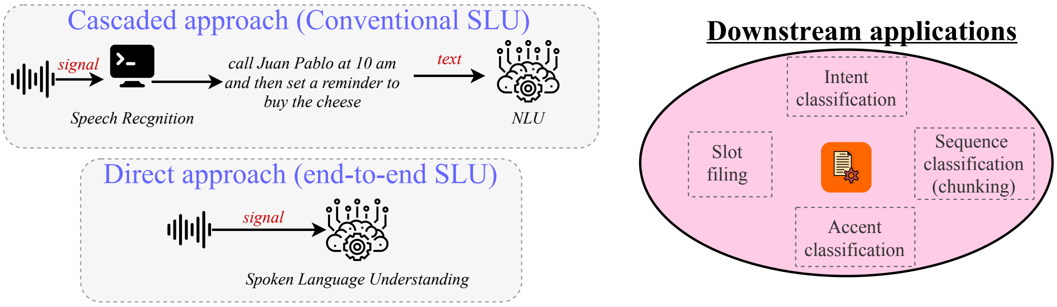

Cascaded approach (Conventional SLU)
signal
call Juan Pablo at 10 am and then set a reminder to buy the cheese
text
Speech Recgnition
NLU
Direct approach (end-to-end SLU)
signal
Spoken Language Understanding
Downstream applications
Intent classification
Slot filing
Sequence classification (chunking)
Accent classification

Figure 2.3: Cascaded and end-to-end spoken language understanding systems. On the right are the main applications covered in this thesis.

example of such models is in Figure 2.2 (c). AED and CTC-based architectures differ from hybrid-based models. For instance, they do not require an explicit LM, as all the blocks are learned end-to-end. However, these E2E models still lag on edge cases, such as recognition of keywords or named entities [15]. Prior work aimed at integrating pretrained LMs intro the decoding frameworks via shallow fusion [69] for improved WER. Intending to overcome the misalignment problem in CTC systems (§ 2.1.2), latest research has targeted a combination of CTC and AED loss functions, termed hybrid CTC/attention [70]. This novel approach has shown improved performance in ASR [71] or in speech-to-text translation [72, 8].

Self-supervised learning Current state-of-the-art (SOTA) models on ASR exploit the self-supervised learning (SSL) paradigm [73, 74, 48]. SSL is a training technique capable of leveraging large-scale unlabeled audio to develop robust large foundational speech models (FSM) [49, 75, 76]. In [77], authors explore a way to perform ASR without using any labeled data in a complete unsupervised fashion. Normally, a fine-tuning stage is required to specialize an SSL-based FSM in a given task. By default, this setup requires much fewer labeled samples compared to standard supervised learning. By applying SSL, these systems have dramatically improved ASR performances on English [49] and multiple other languages [51, 50, 78]. Including models that can perform ASR on more than 1000 languages [79, 80].

2.2 Spoken Language Understanding

Spoken Language Understanding (SLU) is the underlying key component of interactive smart devices such as voice assistants, social bots, and intelligent home devices. Typically, SLU aims at parsing spoken utterances into corresponding structured semantic concepts through a pipeline or “cascaded" approach. Both approaches are listed in Figure 2.3. Effectively interpreting human interactions through classification of intent and slot filling [27] plays a crucial role in SLU, that is why this task has received substantial attention in industry and academia. This thesis covers multiple SLU downstream tasks that can be interfaced from ASR transcriptions or that can be performed in end-to-end fashion.

2.2.1 Cascaded Spoken Language Understanding

In cascaded SLU, the spoken utterances are transcribed by an ASR engine, while its hypotheses are processed by an NLU module, e.g., to identify intents and perform slot filling.[6] The cascaded pipeline is the basic method to perform NLU from speech, i.e., SLU. One main advantage over other approaches is that we do not need paired speech-intent (or slot) samples, as the tasks are carried out by separately optimized models. However, there are some key disadvantages, such as: (1) errors in the ASR transcripts are directly propagated to the NLU module, normally trained only on correct transcriptions; (2) prosodical and non-phonetic aspects present in the spoken utterance are not taken into account. Even though, the classical text-based approach is mostly used in industrial applications and is still an active research area [81].

2.2.2 End-to-end Spoken Language Understanding

More recently, end-to-end (E2E) SLU systems have gained popularity [82, 83, 84, 85]. E2E SLU acts as an individual single model that directly predicts the intent from speech without exploiting an intermediate text representation. In particular, it directly optimizes the performance metrics of SLU. Due to the complex structure of speech signals, a large SLU database along with high-end computational resources (e.g., GPU cluster) are required for training E2E models. In [85], several E2E SLU encoder-decoder solutions are investigated. For instance, instead of directly mapping speech to SLU target [83], pretrained acoustic and language models can be used for downstream SLU tasks, showing to be an effective paradigm [86, 82] over training from scratch.

2.2.3 Intent Classification & Slot Filling

Intent classification Intent classification is a key component of SLU systems, particularly in the context of chatbots, virtual assistants, and other conversational AI applications. Its goal is to determine the intention behind a user's spoken input or query. For example, if a user says, "*Book a flight to Colombia for next Tuesday*" the intent might be: `"book a flight"`. In cascaded approaches, the ASR component can be modeled by fine-tuned SSL-based models such as XLSR [50] or wav2vec 2.0 [49]. In addition, the NLU component can be modeled by well-known pretrained LMs such as, BERT [87], RoBERTa [88], or DeBERTa [89]. The most classical example to perform intent classification follows Equation 2.9.

$$y^i = softmax(\boldsymbol{W^i h_1} + b^i), \tag{2.9}$$

where $\boldsymbol{W^i}$ are learnable weights (of a classification model), $\boldsymbol{h_1}$ is the hidden representations from the `[CLS]` token[7] in BERT, and $\boldsymbol{b^i}$ the bias vector. This is adapted from [90].

[6] In practice, the 1-best transcript representation is the one sent to the NLU model for intent detection.

[7] `[CLS]` represents sentence level classification, and it captures a representation of a whole sentence in a single vector.

Slot filling Slot filling is another important task in NLU, particularly in the context of understanding user queries or commands that involve specific parameters or entities. In slot filling, the goal is to identify and extract relevant pieces of information (slots) from the user's input. These slots typically correspond to specific entities such as dates, times, locations, names, etc. Continuing with the example above, slot filling would involve identifying entities like "`Colombia`" (destination) and "`next Tuesday`" (date of travel). Slot filling helps in extracting structured information from unstructured text, which can then be used by downstream. Equation 2.10 can be used to perform slot filling with a fine-tuned BERT model, as:

$$y_n^s = softmax(\boldsymbol{W^s h_n} + b^s), n \in 1, ..., N. \tag{2.10}$$

Differently from Equation 2.9, in Equation 2.10, we take the remaining hidden states $h_n, n \in 1, ..., N$ instead of only h_1 to classify each input token, see further details in [90].

Other downstream applications In this work, we also propose downstream applications from speech that can be cataloged as SLU, see right block in Figure 2.3 for examples. Other SLU/NLU tasks that are partially covered in this thesis are part-of-speech tagging, chunking, NER [91, 92], and semantic role labeling [93]. As part of this thesis, we propose (1) end-to-end cross-talk and speaker change detection [8]; (2) accent classification for different languages [20]; (3) end-to-end ASR and NER.

2.3 Target Domains and Databases

In this section, we examine the main applications targeted in this thesis. We explore several challenging applications bounded by lack of annotated data or audio quality. The covered applications are in Figure 2.4 to Figure 2.7.

2.3.1 Read & Prompted Speech

Prompted and read speech entails verbal communication where the speaker reads aloud text from a prepared document or reacts to cues provided by an external source, such as a script or instructions. Important databases in this domain include LibriSpeech [94], GigaSpeech [95], TED-LIUM-3 [96], and CommonVoice [97]. Compared to conversational speech, this domain typically employs more formal language. Speakers may adjust their speech pace to synchronize with prompts or scripted text, resulting in a more consistent speech rate. Instances of read speech include audiobook narration or individuals reading from a teleprompter during live broadcasts for prompted speech. In Figure 2.4, we outline the essential characteristics of this speech domain. For further exploration of related work and baseline models, we direct readers to hybrid and E2E-based ASR from Section 2.1. Finally, an important aspect of read and prompted speech is that ASR systems show lower WERs [94, 97] w.r.t conversational speech [98]. See further information in Section 3 of XLSR-Transducer paper [51].

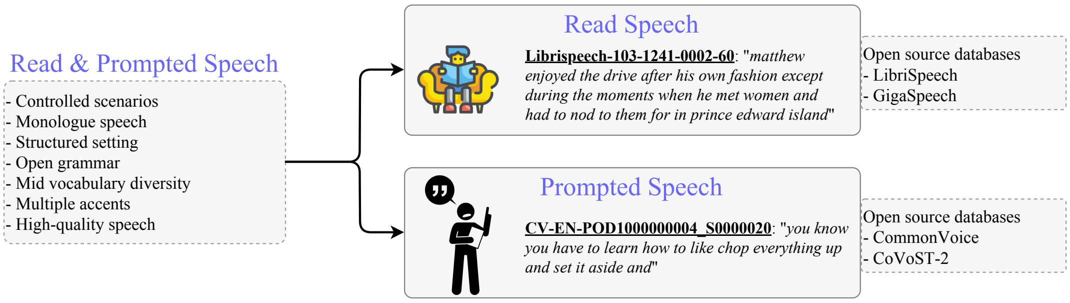

Read & Prompted Speech
- Controlled scenarios
- Monologue speech
- Structured setting
- Open grammar
- Mid vocabulary diversity
- Multiple accents
- High-quality speech
Read Speech
Librispeech-103-1241-0002-60: "matthew enjoyed the drive after his own fashion except during the moments when he met women and had to nod to them for in prince edward island"
Open source databases
- LibriSpeech
- GigaSpeech
Prompted Speech
CV-EN-POD1000000004_S0000020: "you know you have to learn how to like chop everything up and set it aside and"
Open source databases
- CommonVoice
- CoVoST-2

Figure 2.4: Characteristics of read and prompted speech.

Read and Prompted Databases

Librispeech LibriSpeech [94] is a popular ASR benchmark derived from audiobooks, specifically the LibriVox project. The corpus contains approximately 1000h of 16kHz read English speech, and it provides utterance level segmentation. Librispeech is positioned as one of the main datasets to refer when new ASR architectures are developed. In this thesis, we employ the full dataset of 960h and the official dev-{clean,other} and test-{clean,other} subsets.

CommonVoice The CommonVoice dataset is a multilingual corpus of read speech, comprising several thousand hours of audio in more than 100 languages [97]. We use this database across this thesis for multiple task, e.g., ASR and speech-to-text translation. Per language statistics are in Table 2.1. CommonVoice is one of our preferred datasets as it offers: (1) a well-established train/dev/test subsets partitioning; (2) proper and robust annotation protocol for multiple languages; (3) large speaker variability; (4) from low- (Czech or Swedish) to high-resource (English) subsets; (5) it is an evolving dataset (i.e., updated each three months with new data).

CoVoST-2 This dataset targets speech-to-text translation (ST) based on CommonVoice. CoVoST-2 [2] provides data for translating from English into 15 languages (En → X) and from 21 languages into English (X → En). We redirect the reader to the official paper for further details and subset partitioning per language [2].[8]

TED-LIUM3 TED-LIUM v3 consists of audio recordings and transcriptions of TED talks, which are presentations delivered at TED conferences covering a wide range of topics such as science, technology, entertainment, and global issues. In this work, we use the version 3, i.e, TED-LIUM3, which includes the previous two versions [99]. It contains 452h of audio sampled at 16kHz [96].[9] Each recording is composed of a sphere `(.sph)` formatted audio file, and its corresponding transcripts in stm `(.stm)` format.[10]

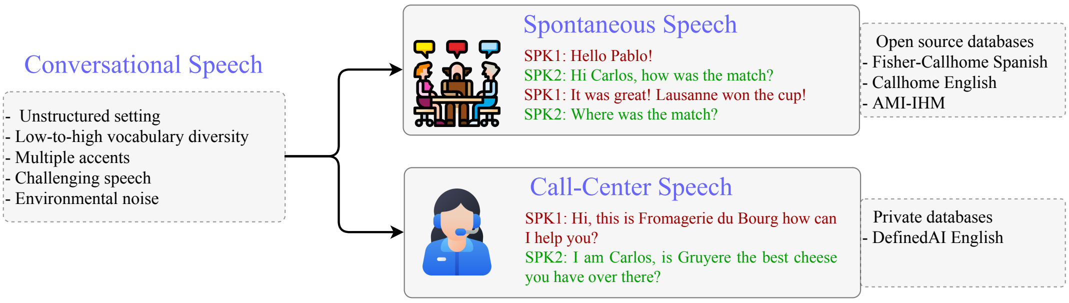
[8] The dataset is also available at: https://github.com/facebookresearch/covost.

[9] The dataset contains 317h/135h male/female audio, with 2028 unique speakers.

[10] The authors utilized the Kaldi Toolkit [100] to align .stm and .sph files.

Table 2.1: CommonVoice train and test splits used for multiple purposes in this Thesis. We use CommonVoice-v11, tag: *cv-corpus-11.0-2022-09-21*.

Language	**Train set**		**Test set**	
	Nb. Utt	**Duration [hr]**	**Nb. Utt**	**Duration [hr]**
English (EN)	947k	1503	16K	27
Catalan (CA)	904k	1403	16.3	28
French (FR)	484k	698	16K	26
German (DE)	478k	759	16K	27
Belorussian (BE)	346k	470	15.8k	26
Spanish (ES)	230k	340	15.5K	26
Italian (IT)	152k	223	15k	26
Dutch (NL)	30k	37	10k	14
Portuguese (PT)	18k	20	8.6k	11
Polish (PL)	16k	24	8.2k	11



SPK1: Hello Pablo!
SPK2: Hi Carlos, how was the match?
SPK1: It was great! Lausanne won the cup!
SPK2: Where was the match?

SPK1: Hi, this is Fromagerie du Bourg how can I help you?
SPK2: I am Carlos, is Gruyere the best cheese you have over there?

Figure 2.5: Characteristics of spontaneous and call-center based conversational speech.

2.3.2 Conversational Speech

In this thesis, we partly focus on conversational speech as it presents increased challenge w.r.t to read and prompted speech. Below, we discuss two types of conversational speech: (1) Spontaneous speech and (2) Call-Center Speech. See Figure 2.5.

Spontaneous Conversational Speech

Spontaneous conversational speech refers to natural, unscripted communication between two or more speakers engaged in a conversation. Unlike prompted or read speech, which may follow a prepared script or respond to external cues, spontaneous conversational speech arises from the spontaneous interaction between speakers without prior planning or rehearsal. It includes everyday interactions, such as informal discussions and casual exchanges in various social contexts. The content of the conversation emerges dynamically based on the participants' thoughts, feelings, and interactions. In addition, it often involves colloquial language, slang, informal expressions, and filler words (e.g., "um," "uh," "like"). Participants take turns speaking and listening, with interruptions and speech overlaps, which transform the task of ASR particularly challenging. Also, the topics can shift regularly within the conversation, reflecting the dynamic nature of conversational speech. In the community, challenging conversational scenarios include: (1) multi-party dialogues [101, 102, 103, 104]; (2) cross-talk and overlapped speech [105, 106, 107]; or (3) conversational speech for specific domains, such as call-center [108, 109] or air traffic control dialogues [4, 5].

Call-center Conversational Speech

Call-center speech is a branch of conversational speech that refers to spoken interactions between call-center agents and customers during customer service or support calls. The conversations are scripted or semi-scripted, and agents are trained to follow specific protocols and guidelines. They focus on addressing customer inquiries, resolving issues, or providing assistance, i.e., rather than engaging in casual conversation. Call-center agents also manage a more formal vocabulary than customers. Overall, conversational speech can be seen as a task-oriented dialogue.

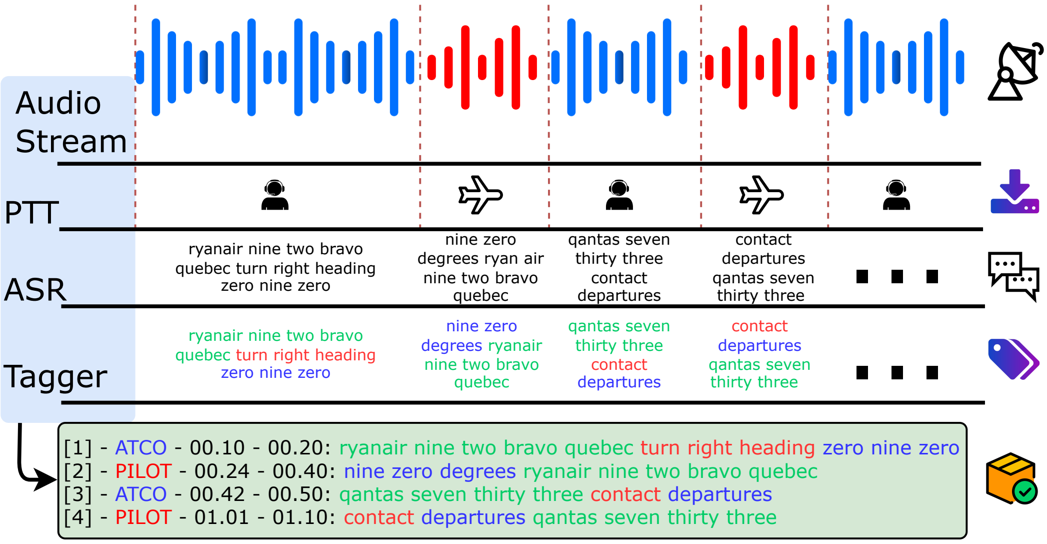
The main challenges when working with call-center audio are: (1) variable noise, where low SNR levels can make the ASR and SLU tasks challenging; (2) users' accent that cannot be known a-priori and might be out-of-domain for the ASR; (3) specific domain and vocabulary. Using ASR and SLU tools within the call-center domain can help to automate pipelines, which can significantly reduce the time spent by agents with customers. This translates to large cost reductions and higher client satisfaction.

Conversational Speech Databases

CallHome English public database The CallHome English dataset (LDC97S42) contains natural conversational stereo-audios between multiple speakers. It consists of 120 unscripted 30-minute telephone conversations between native speakers of English. All calls originated in North America. Most conversations are calls between family members or close friends. The transcript includes named entities annotation. This dataset poses challenges due to its natural conversational nature, known to be challenging for ASR modeling.

Fisher-CallHome Spanish-to-Eglish public database The Fisher and Callhome corpora respectively comprises 170h and 20h of audio and transcripts of telephone conversations in Spanish.[11] The Spanish-to-English translations are available from [98]. We refer to these corpora as Fisher-Callhome. This corpus is well suited for multi-turn ASR and ST, as it contains a significant amount of labeled data and non-segmented (audio) long conversation between speakers. We merged both corpora for training.

[11] Linguistic Data Consortium (LDC) IDs are: LDC2010S01, LDC2010T04, LDC96S35, LDC96T17


Audio Stream
PTT
ASR
Tagger
ryanair nine two bravo quebec turn right heading zero nine zero
nine zero degrees ryan air nine two bravo quebec
qantas seven thirty three contact departures
contact departures qantas seven thirty three
■ ■ ■
ryanair nine two bravo quebec turn right heading zero nine zero
nine zero degrees ryanair nine two bravo quebec
qantas seven thirty three contact departures
contact departures qantas seven thirty three
■ ■ ■

[1] - ATCO - 00.10 - 00.20: ryanair nine two bravo quebec turn right heading zero nine zero
[2] - PILOT - 00.24 - 00.40: nine zero degrees ryanair nine two bravo quebec
[3] - ATCO - 00.42 - 00.50: qantas seven thirty three contact departures
[4] - PILOT - 01.01 - 01.10: contact departures qantas seven thirty three

Figure 2.6: Detailed cascaded pipeline for transcription, tagging, and extraction of key information in an ATCo-pilot conversation. PTT: push-to-talk.

AMI Meeting Corpus AMI is a multi-modal dataset comprising 100h of recorded meetings. For a comprehensive introduction to the corpus, please refer to the corpus overview [110].[12] Approximately two-thirds of the data has been collected using a scenario where participants assume various roles within a design team, progressing through a design project from inception to completion over a day. The remaining portion consists of spontaneously occurring meetings across diverse domains. We only employ the Independent headset microphone (IHM) subset in this thesis.

2.3.3 Air Traffic Control Communications

Air traffic control (ATC) is a service provided by air navigation service providers (ANSPs) with the aim to plan and manage air traffic throughout voice communications. The communication is mainly carried by air traffic controllers (ATCos) and pilots. ATC ensures safe, orderly, and efficient air traffic flow. The primary objectives of ATCos are to prevent collisions between aircraft, maintain safe distances between them, and provide navigational assistance and guidance to pilots [18]. The ATC task has shown to be extremely stressful and highly voice demanding because of the impact a small mistake can make. Several attempts towards increasing the confidence and reducing the workload of pilot-controller communication have been pursued in the past, including experiments with ASR and SLU.[13] In practice, spoken ATC communications are automatically transcribed and then analyzed with SLU systems, typically in a cascaded format. An example of this process is given in Figure 2.6.

[12] Instructions for accessing the data are provided in: https://groups.inf.ed.ac.uk/ami/corpus/.

[13] For example, reducing the amount of time that ATCos spent introducing spoken commands in their workstations.

ATC speech presents multiple challenges w.r.t other domains, e.g., conversational or read speech. See Figure 2.7 for further information. This thesis explains how we can overcome those challenges to build useful tools, e.g., reducing ATCo's workload[14] by integrating different ASR and SLU systems [4, 5, 18] in a cascaded format [111].

Challenges and Motivations

The ATC domain can be catalogued as a low-resource constrained and challenging scenario that presents several challenges, as shown below.

Signal perspective ATC communications present unique challenges from a signal processing perspective, particularly due to the noisy nature of the speech compared to standard ASR corpora. Pilot communications are typically transmitted via very-high-frequency (VHF) receivers that introduce both channel and cockpit noise, with signal-to-noise ratios (SNR) ranging from 5 to 20 dB. In this thesis, we leverage data collected from multiple sources that employ low-cost hardware to gather extensive quantities of ATC communications near airports. Although this data is typically unannotated and of lower quality, it provides a significant resource for developing robust ASR systems by integrating additional contextual information, such as radar data, to compensate for audio quality deficiencies. Similarly, obtaining high quality data from ANSPs is very difficult, for legal issues [23]. It is important to note that ASR systems that lack training on enhanced data or clean speech inputs, such as those from ANSPs, may produce transcripts that are too noisy for effective use in subsequent SLU systems.

From a signal perspective, working with ATC speech introduces several critical challenges:

- High variability in speech due to factors like stress and fatigue among speakers;
- speech variations between different speakers;
- wide range of accents and dialects;
- the unique nature of ATC communications, which do not fit into categories of spontaneous, read, or command speech.

Addressing these issues is crucial for advancing the reliability and accuracy of ASR systems for the ATC domain.

Data scarcity A limitation in developing highly-accurate ASR and SLU systems for ATC is the lack of supervised data. Likewise, generate the transcriptions of such data is extremely costly, e.g., a raw ATCo-pilot voice communication recording of one hour–including silences–requires between eight and ten man-hours of transcription effort [112].[15] This produces ~ 10 to 15 minutes of pure speech, after removing the silences [113, 112].

[14] Note that workload reduction might translate to reduced flight time, e.g., decreasing overall operational costs and the environmental impact of aircraft.

[15] Mainly as it requires highly trained participants, often active or retired ATCos

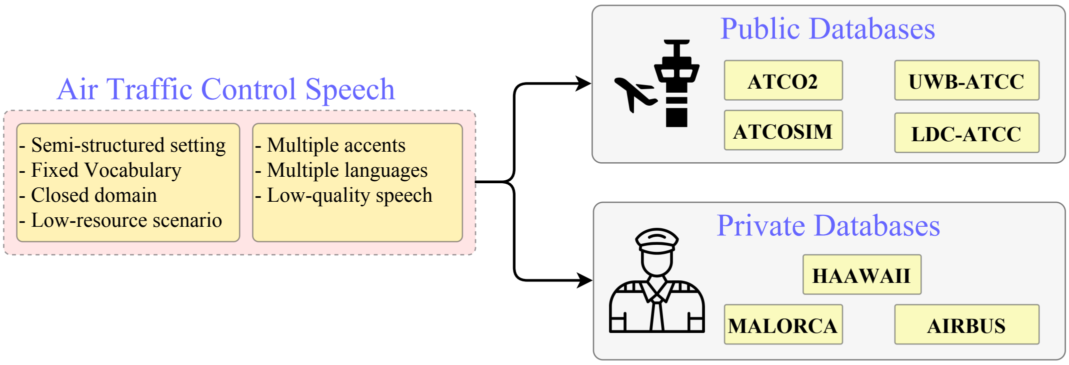


Figure 2.7: Characteristics of air traffic control communications speech.

Constrained domain ATC is built for specific domains, e.g., one airport or en-route/approach scenarios [18]. The process of adapting pretrained ASR and SLU models to different airports or control areas requires new in-domain data, which remain challenging to collect and annotate. For instance, ATC audio data collected from one airport, e.g., `airport` X, in general, do not transfer well to `airport` Y.

Related Work

Research attempting to aid ATCos by ASR dates back as far as the 70s'. First, systems aimed at isolated word recognition, speaker verification and command recognition for military applications [126]. Exploratory research towards integration of ASR technologies to aid ATCos started in the late 80s, with [127]. Several other research directions target user-friendly and robust automatic systems to train ATCos, or the so called 'pseudo-pilots' [128]. Akin training systems have been proposed by [129, 113, 130, 131, 124].

We shortlist 4 well-known European-based projects that aims at developing speech and text-based tools to aid ATCos in their daily tasks. Initially, *MALORCA* project[16] was a step forward in demonstrating that ASR tools can cut down ATCos workload [132] while increasing the overall efficiency [133]. Then, *HAAWAII* project[17] has led initiatives to extract key entities (e.g., named-entity recognition or slot filling) in the transcribed dialogues produced by an ASR system [134]. *SESAR2020's Solution 97.2* [135] stands as one of the first attempts to analyze the impact of ASR and SLU tools in the performance of ATCos in simulated tower and ground environments. Finally, *ATCO2* project (our corpora) aimed at reducing the human work needed to develop ASR and SLU tools for ATC, mainly by integrating semi-supervised techniques to improve the pseudo-transcription process [22, 18]. While the *MALORCA*, *HAAWAII* and *Solution 97.2* corpora are not public, *ATCO2* developed a pipeline to collect large quantities of ATC speech data, which are distributed to the public through ELDA.[18]

[16] MAchine Learning Of speech Recognition models for Controller Assistance: http://www.malorca-project.de/wp/.
[17] Highly Automated ATC Workstations with Artificial Intelligence Integration: https://www.haawaii.de/wp/.
[18] Available for purchase at: http://catalog.elra.info/en-us/repository/browse/ELRA-S0484.

Table 2.2: Air traffic control communications related public and private ATC databases. The *ATCO2 corpus* is a large-scale public database with audio, pseudo-labels and radar information. †full database after silence removal. ††speaker accents depend on the airport's location, however, the accent of pilots are not known at any time of the communication due to privacy regulations. Table taken from previous work [5].

Database	**Details**	**Licensed**	**Accents**	**Hours†**	**Reference**
		Private databases			
HAAWAII	Real data from Iceland and London airports	✗	Icelandic, British	47	[13]
MALORCA	Real data: LOWW and LKPR	✗	German, Czech	13	[114] [115]
AIRBUS	Real data from LFBO	✗	French	100	[116]
VOCALISE	Real data from terminal maneuvering area and area control center in France	✗	French	150	[117]
ENAC	Real data from two French en-route control centers and one major airport	✗	French	22	[118]
		Public databases			
ATCOSIM	Simulated in studio, added cockpit noise. Recordings split by gender (Male/Female)	✓	Swiss German, German, French	10.7	[119]
UWB-ATCC	Real data from LKPR	✓	Czech	13.2	[120]
LDC-ATCC	Real data from 3 US airports: KBOS, KDCA and KDFW	✓	American English	26.2	[121]
HIWIRE	Simulated in studio, ATC prompts, added cockpit noise	✓	French, Greek, Italian, Spanish	28.7	[122]
ATCSpeech	Real accented Mandarin Chinese and English	✓	Chinese and English	57.8	[123, 124]
		Corpora released by ATCO2 project			
ATCO2 corpora	ATC data from different airports and countries. Low quality but large-scale data.				
ATCO2-test-set	Transcribed audio	✓	Several††	4	
ATCO2-PL-set	Pseudo-transcribed audio	✓	Several††	5281	[22]
		Free access databases releseased by ATCO2 project			
ATCO2-test-set-1h	'ASR dataset': https://www.atco2.org/data	✓	Several††	1	[22]
ATCO2-ELD set	'LID dataset': https://www.atco2.org/data	✓	Several††	26.5	[125]

Public and Private ATC corpora For our experiments, we use all the open-source corpora from Table 2.2, except *ATCSpeech*. The data provided by ANSPs is much higher quality than the one collected by ATCO2. This is due to the hardware quality used to record the audio data. In addition to these resources, as we have access to private corpora, we use them for supervised training. Specifically, we use *HAAWAII*, *MALORCA*, and *AIRBUS*. For brevity, we do not cover the details of each database. We redirect the reader to published work in [18, 5, 22, 21] for further details.

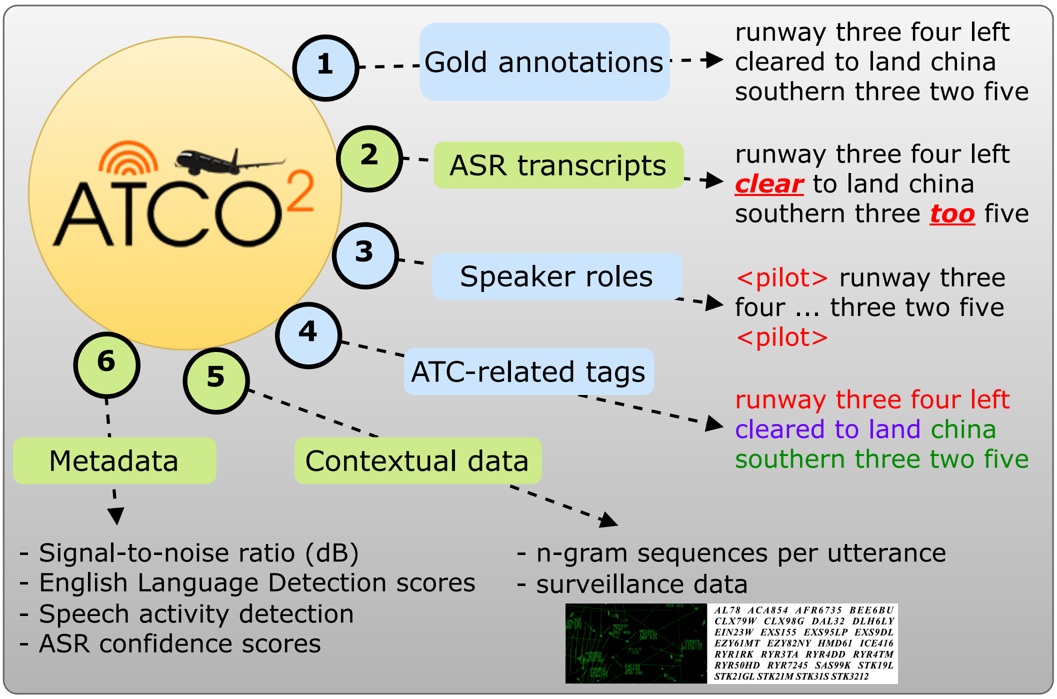


Figure 2.8: ATCO2 corpus ecosystem. Blue circles denote transcriptions only available for *ATCO2 test set corpus*. Green circles denote transcriptions and metadata available for both *ATCO2 test set* corpus and *ATCO2 pseudo-labeled* corpus (see Table 2.2 bottom).

Public corpora released - ATCO2: the ATCO2 corpora[19] is one of our main test-beds for ASR and SLU systems for ATC. In this setup, we have access to large-scale databases, albeit usually of lower quality as the ones provided by ANSPs. The ATCO2 provides pseudo-labeled ATC data for more than 10 airports, see Table 2.2. An important part of the work on ATC includes the recording, processing, collection, and labeling (or pseudo-labeling) of the ATCO2 corpora. See further details in Figure 2.8. In [5] we cover technical details about how we collected, prepared and open-source this data for the wide research community.[20]

2.4 Evaluation Metrics

In this thesis we validate results with a wide range of evaluation metrics. This includes acoustic and text-based metrics, as shown below.

[19] ATCO2 project website: https://www.atco2.org/.

[20] The collection, pre-processing and pseudo-labeling of the ATCO2 corpora is a significant contribution of this thesis. See the corpus open-sourced at https://catalogue.elra.info/en-us/repository/browse/ELRA-S0484/.

2.4.1 Metrics for Speech-based Models

Word error rate For ASR, we employ the word error rate (WER) metric. Given a reference transcript and a hypothesis produced by an ASR system, we can compute WER with Equation 2.11.

$$WER\,[\%] = \frac{\text{Insertions + Deletions + Substituticions}}{\text{Total Number of Words in Reference}}, \tag{2.11}$$

where the number of insertions, deletions, and substitutions are obtained by aligning reference and hypothesis with edit distance [100]. WER is measured in percentage and lower WER means more accurate ASR system.

Character error rate Similar to WER, character error rate (CER) focuses on errors produced by the model only on the character level. Before running edit distance, we split the reference and hypothesis in characters and then compute CER. Phoneme error rate (PER) is also a widely used metric, though is not employed in this thesis. CER is useful (over WER) when a more granular analysis of errors is required. It therefore provides insights into the accuracy of individual characters recognized by the ASR system.

Real-time factor & Latency Throughput is measured using the inverse of the real-time factor (RTF) metric. It is defined such as:

$$RTFX = \frac{audio\ inferred\ (seconds)}{compute\ time\ (seconds)}, \tag{2.12}$$

it is the inverse of the RTF (Real Time Factor) metric, such as RTFX = 1/RTF.

Acoustic-based diarization For speaker diarization (SD), we use diarization error rate (DER) and Jaccard Error Rate (JER) as metrics. DER measures the fraction of time that the segment is not attributed correctly to a speaker or to non-speech, as shown below:

$$DER = \frac{\text{false alarm} + \text{miss detection} + \text{speaker confusion}}{\text{Total duration of speech in the reference file}}, \tag{2.13}$$

where false alarm is the duration of non-speech incorrectly classified as speech, missed detection is the duration of speech incorrectly classified as non-speech, confusion is the duration of speaker confusion, and total is the total duration of speech in the reference. JER is a recently proposed metric [136] that avoids the bias towards the dominant speaker, i.e., evaluating equally all speakers. The JER is defined in Equation 2.14:

$$JER = 1 - \frac{1}{\text{\#speakers}} \sum_{\text{speaker}} \text{max}_{\text{cluster}} \frac{|\text{speaker} \cap \text{cluster}|}{|\text{speaker} \cup \text{cluster}|}, \tag{2.14}$$

where speaker is the selected speaker from reference and *$max_{cluster}$* is the cluster from the system with maximum overlap duration with the currently selected speaker.

2.4.2 Metrics for Text-based Models

Named-entity WER & Accuracy In addition to WER, we evaluate the accuracy and WER of ASR systems only on named entities (NEs): *NE-A* and *NE-WER*. Both metrics are calculated after the reference and hypothesis are aligned, and only on the strings containing NEs. NE-A is computed binary, i.e., "correct" – when the NE is completely recognized correctly, "incorrect" – when at least one error occurs within the NE. An example of this metric for ATC systems is the Callsign-WER, which measueres WER only on the sequence of words that belong to a callsign in an utterance.

Out-of-vocabulary ratio In ASR systems, out-of-vocabulary (OOV) words refer to words that are not present–thus cannot be hypothesized–in the system's lexicon. When an ASR system encounters an OOV word at decoding time, it may struggle to accurately recognize and transcribe it, leading to higher WERs overall. For instance, OOV words are categorized with the `"<unk>"` (unknown) symbol, depending on the system's design. The OOV ratio can be computed as the total number of OOV words divided by the total number of words.

BLEU score BLEU is a method for evaluation of machine translation (and speech-to-text translation) systems. BLEU score was introduced in [137] and is composed by:

1. **Modified Precision**:
   - Used to measure the accuracy of n-grams (contiguous sequences of n words) in the candidate translation w.r.t the reference translations;
   - precision for each n-gram length (from 1 to N) is computed and then averaged using geometric mean.
2. **Brevity Penalty (BP)**:
   - The brevity penalty factor (BP) is applied to account for the length of the candidate translation compared to the reference translations.
   - If the candidate translation length (c) is shorter than the effective reference corpus length (r), BP is calculated as $e^{(1-r/c)}$, else $BP = 1$.
3. **Combining Precision and Brevity Penalty**:
   - The BLEU score is computed by multiplying the brevity penalty (BP) with the exponential of the weighted sum of the logarithms of the modified precision scores for all n-gram lengths up to N.
   - The weights w_n are positive values that sum up to 1 and are typically uniform (e.g., $w_n = \frac{1}{N}$ for N equal to 4).

The BLEU score is designed to capture the precision of the candidate translation relative to the reference translations, while also considering brevity, as represented by Equation 2.15. A higher

BLEU score indicates a better match between the candidate and reference translations [137].

$$\text{BLEU} = \text{BP} \times \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{2.15}$$

3 Data and Task Bounded Low-Resource Speech Recognition

Introduction

In this chapter, we propose multiple approaches on how to develop high-performance ASR systems for challenging low-resource applications such as air traffic control (ATC) (§ 3.1) and conversational speech (§ 3.4). Additionally, we propose strategies on how to bypass challenges due to amount of supervised data (§ 3.2) or how to leverage contextual information at decoding (§ 3.3 and § 3.4) and training time (§ 3.5). An graphical overview of this chapter is in Figure 3.1.

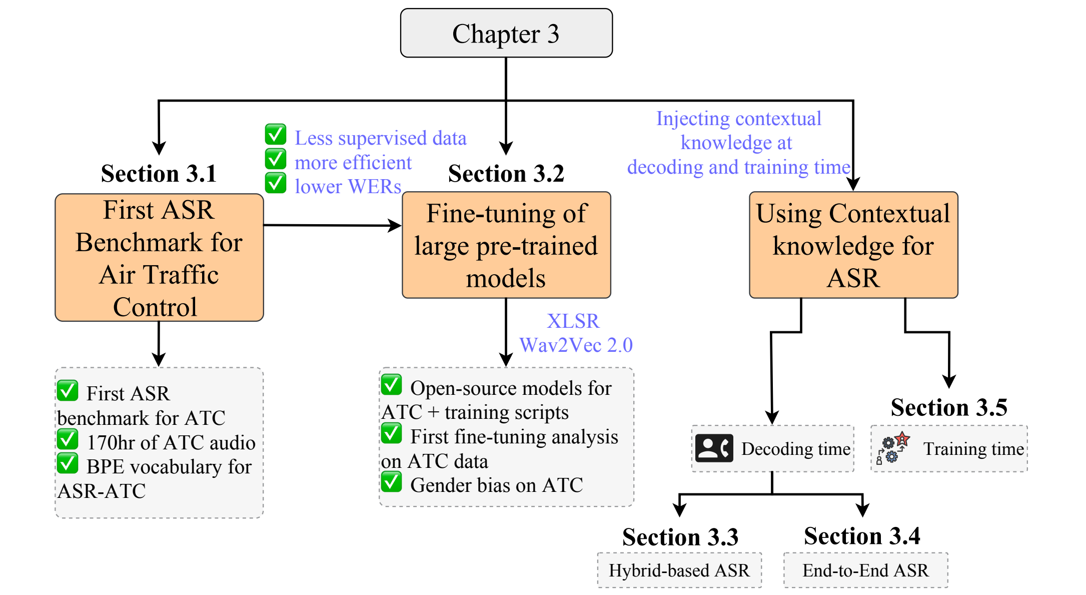

Chapter 3
Section 3.1
First ASR Benchmark for Air Traffic Control
✓ Less supervised data
✓ more efficient
✓ lower WERs
Section 3.2
Fine-tuning of large pre-trained models
Injecting contextual knowledge at decoding and training time
Using Contextual knowledge for ASR
✓ First ASR benchmark for ATC
✓ 170hr of ATC audio
✓ BPE vocabulary for ASR-ATC
XLSR
Wav2Vec 2.0
✓ Open-source models for ATC + training scripts
✓ First fine-tuning analysis on ATC data
✓ Gender bias on ATC
Decoding time
Section 3.5
Training time
Section 3.3
Hybrid-based ASR
Section 3.4
End-to-End ASR

Figure 3.1: Overview of Chapter 3.

3.1 Supervised ASR Learning for Challenging Applications

This work conveys an exploratory benchmark of several state-of-the-art ASR models trained on more than 170 hr of air traffic control (ATC) speech. We demonstrate that the cross-accent flaws due to speakers' accents are minimized when we scale up the supervised training data, making the system suitable for the challenging ATC domain. To the author's knowledge, this is the first time that such an amount of ATC-related databases have been employed to developed ASR systems. Our ASR system attains a WER of 7.75% across four databases. An additional 35% relative improvement in WER is achieved when training a TDNNF system with BPE based vocabulary.

Publication Note

The material presented in this section has been used in the following publications:

- J. Zuluaga-Gomez, P. Motlicek, Q. Zhan, K. Veselý, and R. Braun, "Automatic Speech Recognition Benchmark for Air-Traffic Communications," in *Proc. Interspeech*, 2020, pp. 2297–2301
- J. Zuluaga-Gomez, I. Nigmatulina, A. Prasad, P. Motlicek, D. Khalil, S. Madikeri, A. Tart, I. Szoke, V. Lenders, M. Rigault, *et al.*, "Lessons Learned in Transcribing 5000 h of Air Traffic Control Communications for Robust Automatic Speech Understanding," *Aerospace*, vol. 10, no. 10, p. 898, 2023
- J. Zuluaga-Gomez, A. Prasad, I. Nigmatulina, P. Motlicek, and M. Kleinert, "A Virtual Simulation-Pilot Agent for Training of Air Traffic Controllers," *Aerospace*, vol. 10, no. 5, p. 490, 2023

Supplementary materials related to this section:

- **Code**: GitGub repository at https://github.com/idiap/atco2-corpus
- ATCO2 project website: https://www.atco2.org/

Major contributions Problem definition and experimental design and setup. Data preparation. Trained the hybrid-based ASR systems for the experiments. Lead the work, including the paper write up.

3.1.1 Introduction

The communication methods between pilots and Air-Traffic Controllers (ATCos) have remained almost unchanged for many decades, where the ATCo's main task is to transfer spoken guidance to pilots during all flight phases (e.g., approach, landing, or taxi) and at the same time providing safety, reliability, and efficiency. This task has shown to be extremely stressful because of the consequences that a small mistake can generate. Several attempts towards increasing the confidence and reducing the workload of pilot-ATCo communication have been pursued in the past, including experiments with ASR. Initially, due to budget and scarcity of computing power, previous work only targeted isolated word recognition, or 'voice activity detection'. Currently, most research targets low-latency ASR. Military applications were one of the first attempts involving engines for command-related ASR; Beek et al. [126] studies ASR within military applications such as speaker verification, commands recognition and system control of aircraft. They concluded that ATC speech has a very limited vocabulary, speaker-dependent issues and environmental noises that need to be addressed to produce a sufficiently-reliable system. Initially, the integration of ASR technologies in ATCo started in the late 80s' with Hamel et al. report [127]; but lately, ASR technologies has been successfully deployed on ATC training simulators. For example, Matrouf et al. [128] proposed a user-friendly and robust system to train ATCos based on hierarchical frames and history of dialogues, i.e., context-dependent system. Similarly, DLR [128], MITRE [129] and more recently UPM-AENA [113] under the INVOCA project proposed akin training systems. Previous project MALORCA has demonstrated that ASR tools can reduce ATCos workload [132] and increase the efficiency [133], where also it addressed the lack of transcribed ATC speech data using semi-supervised training[1] to decrease WERs and command error rates [114, 115]. ATCO2 project aimed at developing a unique platform to collect, organize and pre-process air-traffic speech data from airspace available either directly through publicly accessible radio frequency channels (such as LiveATC [140]), or indirectly from ANSPs. One of the current challenges of ASR engines for ATCo communications is the changing ATCos accent and vocabularies across different airports.

In this work, we present the first results of ATCO2 based on six ATC corpora. First, we explore transfer learning from a Deep Neural Network (DNN) system trained on an out-of-domain (OOD) corpus, then we contrast the results with SOTA ASR systems, e.g., TDNNF [141] and CNN+TDNNF. Secondly, given the high likelihood of out-of-vocabulary (OOV) words ratio due to the intrinsic changing behavior of the air-space, we experimented with Byte-Pair Encoding (BPE) based vocabularies [67], as it allows the ASR system to recognize words not seen during training (but part of the vocabulary).

3.1.2 Databases and Experimental Setup

Databases The ATC-related databases used in this work are listed in Table 2.2 and in Table 3.1. This accounts for nearly 180 hr (training and test sets) of pure ATC speech. In this work, we also measured the impact of transfer learning from ASR engines trained on out-of-domain databases as part of the proposed benchmark: (1) we merge Librispeech [94] (960h) and Commonvoice [97] (500h English subset); (2) we pre-train a TDNN-F model; and finally, (3) we adapt the pretrained models using in-domain data. See the data description Section 2.3.1. In order to measure the impact of the amount of training data for ASR, we merged six command-related databases in three training sets as shown in Table 3.1. In case of ATCOSIM, we split the database (by speakers) in a 80/20 ratio (i.e., we used 80% of data as train/validation and the remaining 20% as test set). In case of MALORCA database, it comprises two ATC approaches (collected from two ANSPs), Vienna and Prague. The remaining databases were collected, processed and released from different projects; we redirect the reader to their references. In fact, AIRBUS held in 2018 a challenge [116] related to ASR and callsign detection (CSD) of ATCos speech segments; Authors in [142] convey the results of the top 5 teams. It is important to mention that we do not compare our results with theirs because the evaluation set (5h) was not released by AIRBUS; nevertheless, we created from the train data a test set of 350 utterances ($\sim$1h). The proposed acoustic models are evaluated on four different test sets, where characteristics such as ATCo accent, spoken commands, airport's origin and quantity of training data vary.

[1] also studied for under-resourced languages [138, 139]

Lexicon The word list used to build the lexicon was assembled from the transcripts of all the ATC databases (i.e., Tr1+Tr2, in Table 3.1) and from some other publicly available resources (i.e., lists with names of airlines, airports, and ICAO alphabet, see [5]). The pronunciations were synthesized with Phonetisaurus [143]. The G2P (grapheme-to-phoneme) model was trained on Librispeech lexicon, and we inherited its set of phonemes. Additionally, we adopted a BPE-based vocabulary system [67], limiting the number of subword mergers to 2000. BPE efficiently segments words into smaller units, facilitating the handling of an open vocabulary and the integration of new lexical entries, particularly useful in ASR systems as evidenced by various studies [144, 145, 146]. This feature is particularly advantageous for ATC communications, which predominantly use callsigns but also include a significant number of foreign proper nouns potentially absent in conventional word-based models. For generating pronunciations, we utilize a character-based sub-word lexicon where words are decomposed into characters—used in place of phonetic units—to derive their pronunciations.

Table 3.1: ATC in-domain training and test sets. OOD denotes out-of-domain set.

Train data-sets		
Name	**Hours**	**Description**
Train1	38.7	ATCOSIM (train) + MALORCA (Vienna+Prague) + UWB-ATCC
Train2	137.7	AIRBUS + LDC-ATCC + Hiwire
Tr1+Tr2	176.4	Train1 + Train2
OOD set	$\sim$1500	Out-of-domain set: Librispeech + Commonvoice
Test data-sets		
ATCOSIM	2.5	20% of ATCOSIM train set
PRAGUE	2.2	From MALORCA set
VIENNA	1.9	From MALORCA set
AIRBUS	1	From AIRBUS set

Language modeling For language modeling, we employed n-gram LMs developed using SRI-LM [147], training on the transcripts from both Tr1 and Tr2 datasets. Initially, we utilize a 3-gram model (denoted as 'LM-3') for decoding, followed by a 4-gram model ('LM-4') for re-scoring purposes, as shown in our results (Table 3.2). Additionally, a 6-gram model ('LM-6') was developed specifically for our BPE setup.

ASR model training All experiments are conducted using the Kaldi speech recognition toolkit [100]. We report results on two state–of-the-art DNN-based acoustic architectures. We train Factorized TDNN or TDNNF [141] with $\sim$1500h of OOD speech (see Table 3.1) and then we adapt the resulting model with the three proposed training sets, Train1, Train2 and Tr1+Tr2. Afterward, we perform flat-start CNN+TDNNF training without any kind of transfer learning or adaptation; the idea behind this is to measure quantitatively whether the amount/accent of training data helps to reduce WERs. We use the standard chain lattice-free maximum mutual information (LF-MMI) based Kaldi's recipe for both architectures, which includes 3-fold speed perturbation and one third frame subsampling.

Lattice-free maximum mutual information training LF-MMI training of TDNNF models still relies on a HMM-GMM model to build both the alignments and lattices needed during training. The HMM-GMM models are trained with only the OOD, i.e., Librispeech + Commonvoice. We followed the standard Kaldi's recipe which requires 100-dimensional i-vector features, 3-fold speed perturbation, and lattices for LF-MMI training supervision. The TDNNF system trained on the OOD training set ($\sim$1500h) is labeled as 'TDNNF-B'. To measure the impact of the amount of training data on performance in the target domain, we train once with and once without transfer learning on the three different ATC train sets presented in Table 3.1. Models trained with transfer learning have 'TF' in the name, e.g., TDNN-TF-B. The systems without transfer learning simply are denoted according to their architectures, i.e., TDNNF, CNN+TDNNF or TDNNF-BPE.

3.1.3 Results & Discussion

Results of Table 3.2 are split into four blocks. First, TDNNF-B is trained on a 1500h OOD set. This is our base model for transfer learning. Second, TDNNF-B model is adapted to the different ATC datasets (using TDNNF-B as seed), i.e., Train1, Train2 and Tr1+Tr2. Third, we compare WERs for TDNNFs without transfer learning. Finally, we present results on a CNN+TDNNF chain model and TDNNF trained with BPE units. The base model performs poorly on the ATC data. This is not surprising as Librispeech and Commonvoice are both read speech with mostly clean audio. ATC speech is more noisy, the speakers talk much quicker, and the accents are stronger. Despite the significant difference in domains, the pretraining still helps when the target dataset is not too large, i.e., compare first two rows in Table 3.2. Note that large differences in performance of the models trained on either Train1 or Train2 can be explained by whether the accent(s) in the test set were also present in the training set. Once the target domain dataset becomes large enough, we do not see the benefit of pretraining (see the last row of the TDNNF-TF-B and the TDNNF models). The last block of experiments provides a broader cover of different DNN architectures and techniques on our proposed ASR benchmark for ATC communications. There is no clear winner. The CNN+TDNNF system yielded a new baseline of 5% WER for ATCOSIM, showing a relative improvement on WERs of 16.7% and 3.9% when compared to TDNNF-TF-B and TDNNF. The best model for Vienna dataset was TDNNF trained on Tr1+Tr2 and scored with a 4-gram LM, whereas for Prague it was TDNNF with 6-gram and lexicon based on BPE. Compared to previous experiments on MALORCA [114, 115], our approach yields 29.8% and 37.9% relative WER improvement for Vienna and Prague.

We further investigated why the BPE model performs significantly better on the Prague test set, and found that the difference in performance is entirely explained by reduced deletions (five times more deletions in TDNNF and CNN+TDNNF than TDNNF-BPE system). The word-based model is obviously not able to recognize OOV words, which is the primary reason for the deletion errors. The OOV rates on Prague, Vienna, AIRBUS and ATCOSIM test sets are 3.3%, 1.1%, 0.0% and 0.1%. This shows that the BPE system is capable of recognizing OOVs and thereby improving performance; although, it does come at a cost since the BPE models also perform significantly worse on some test sets. Additionally, we noticed that the BPE based model performs better

Table 3.2: ASR benchmark with different ASR architectures, vocabularies, and amount of in-domain and OOD training data.

			Word Error Rates (WER) % - (test sets)							
System	**Train Set**	**Params.**	**Vienna**		**Prague**		**AIRBUS**		**ATCOSIM**	
			LM-3	LM-4	LM-3	LM-4	LM-3	LM-4	LM-3	LM-4
TDNNF-B	OOD set	23.1M	95.8	95.8	47.6	43.3	80.6	77.5	67.5	63.4
TDNNF-TF-B	Train1	20.8M	7.6	7.1	9.1	9.0	53.6	51.4	7.5	7.3
	Train2		30.2	26.2	19.3	17.8	14.9	14.6	23.9	20.5
	Tr1+Tr2		7.5	6.9	8.6	8.4	15.2	14.7	5.9	6.0
TDNNF	Train1	20.8M	8.1	7.5	8.9	8.7	67.8	66.7	8.5	8.1
	Train2		33.2	30.2	20.1	18.8	14.6	14.5	23.4	19.6
	Tr1+Tr2		**7.1**	**6.6**	8.1	7.9	**14.6**	**14.4**	5.3	5.2
CNN+TDNNF	Tr1+Tr2	14.3M	7.1	6.7	8.1	7.9	15.1	14.7	**5.0**	**5.1**
			LM-6 (BPE)		LM-6 (BPE)		LM-6 (BPE)		LM-6 (BPE)	
TDNNF-BPE	Tr1+Tr2	20.8M	7.6		**5.1**		15.1		7.2	

on foreign words (even when the word-based model includes these words). We attribute this to the character-based lexicon system, which generalizes better to foreign languages which are not closely related to English. For the ATCOSIM corpora, we obtain 63.4% WER with TDNNF-B and an improvement to 8.1% WER when training only on Train1 set. An additional 10% relative WER improvement can be obtained if employing transfer learning, reaching 7.3% absolute WER. Finally, with the intention to explore different amount of training data and ASR architectures, we reach an absolute WER of 5.0% when using a CNN+TDNNF system trained on Tr1+Tr2.

Conclusions This work introduces a benchmark of different ASR architectures for ATC speech. This is the first study employing six ATC databases spanning more than 176h of speech data. In addition, these corpora strongly related in both, phraseology and structure to ATCos-pilots communications. Therefore, this work partly deals with the challenge that arises from the lack of supervised databases, that many previous studies have referenced.

3.2 Fine-Tuning of Large Pretrained Models for ASR

Recent work on self-supervised pre-training focus on leveraging large-scale unlabeled speech data to build robust E2E acoustic models (AM) that can be later fine-tuned on ASR. Yet, few works investigated the impact on performance when the data properties substantially differ between the pre-training and fine-tuning phases, termed domain shift. We target this scenario by analyzing the robustness of Wav2Vec 2.0 and XLS-R models on downstream ASR for a completely unseen domain, air traffic control communications.

Our contributions answer the three questions below:

1. How robust pretrained E2E models are on new domains, such as ATC?
2. How much in-domain ATC labeled data is required to fine-tune an E2E model that reaches on-par performance with regard to hybrid-based models?
3. How robust are E2E models on speech from different genders?

Publication Note

The material presented in this section is adapted from the following publication:

- J. Zuluaga-Gomez, A. Prasad, I. Nigmatulina, S. S. Sarfjoo, P. Motlicek, M. Kleinert, H. Helmke, O. Ohneiser, and Q. Zhan, "How Does Pre-trained Wav2Vec 2.0 Perform on Domain-Shifted ASR? An Extensive Benchmark on Air Traffic Control Communications," in *IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 205–212

Supplementary materials related to this section:

- **GitHub repository at:** https://github.com/idiap/w2v2-air-traffic
- ATCO2 project website: https://www.atco2.org/
- Pretrained ATC models in HuggingFace: https://huggingface.co/Jzuluaga/, IDS: *wav2vec2-large-960h-lv60-self-en-atc-atcosim*, *wav2vec2-xls-r-300m-en-atc-uwb-atcc-and-atcosim*, *wav2vec2-large-960h-lv60-self-en-atc-uwb-atcc*, *wav2vec2-xls-r-300m-en-atc-uwb-atcc* and *wav2vec2-xls-r-300m-en-atc-atcosim*

Major contributions Problem definition and experimental design and setup. Data preparation. Training and E2E fine-tuning for ASR experiments. Lead the work, including the paper write up.

3.2.1 Introduction

A substantial amount of recent work on E2E acoustic modeling including ASR exploits self-supervised learning (SSL) of speech representations [48] including autoregressive models [73, 74] and bidirectional models [48, 76]. Self-supervised learning is a training technique capable of leveraging large-scale unlabeled speech to develop robust acoustic models [49, 75]. In fact, [77] explores a way to perform ASR without any labeled data in a complete unsupervised fashion. In a standard setup, E2E models trained by SSL are later fine-tuned on downstream tasks with much fewer labeled samples compared to standard supervised learning. With the use of SSL, the systems have dramatically improved ASR performances on English speech datasets [49], such as LibriSpeech [94]. Similarly, performance on cross-lingual speech recognition is largely improved by SSL [148, 50]. It can be assumed that SSL-based pre-training allows models to capture a good representation of acoustics, which can be leveraged across different languages for ASR. This work reviews the robustness of two well-known E2E acoustic models trained by SSL (i.e., Wav2Vec 2.0 and XLS-R) on a completely unseen domain: air traffic control (ATC) communications.

Contribution and motivation Only a few previous works intended to measure the effect of domain mismatch between pre-training and fine-tuning phases of E2E models [149]. First, we show that E2E SSL pretrained models learn a strong representation of speech. Fine-tuning on a downstream task is computationally less expensive than training from scratch, and it requires less in-domain supervised data to achieve comparable WERs w.r.t hybrid-based ASR. Second, we hypothesize that pretrained multilingual models (i.e., [50]) perform better on ATC speech data that contains accented English. Potentially, due to the strong speech representation acquired during SSL phase, which translates into a more accent-agnostic AM. Third, E2E models have gained exponential interest in the research community. Even so, little investigation has been carried out about estimating the WERs gap produced by gender disparities (few ones [150, 151, 152]). In this work, we fill this gap by analyzing running experiments with ATC audio from different genders.

We believe this work is impactful because the ASR field is advancing in a fast manner, where each month many FSM are poured into the research field with outstanding performances on well-known corpora, e.g., LibriSpeech [94]. Nonetheless, little has been examined in many other domains, such as ATC communications. Thus, it is of particular interest to evaluate and assess the performance of these FSM on *'lagged'* fields.

3.2.2 Databases and Experimental Setup

This research experiments with seven ATC datasets in the English language with various accents, variable speech rate, and data quality, as listed in Table 3.3. With the aim of encouraging open research on ATC,[2] we experiment with four public databases. To the author's knowledge, this is the first work that open sources code in the field of robust ASR targeted to ATC.

Databases For all of our experiments, we up-sample all audio recordings to 16 kHz. Additionally, there are not any official train/dev/test splits for LDC-ATCC, UWB-ATCC and ATCOSIM databases. Therefore, we split them following the proportions in Table 3.3. We also make sure that there is no speaker or utterance overlaps between each subset.

ASR model training Our experimental setup is split into three parts, which aims to answer each of the questions raised in the Section 3.2.1. Initially, we compare the WERs of several E2E models when fine-tuned with ATC audio. We define two training datasets, i) 32h of annotated

[2] Generally, ASR for ATC lags behind due to privacy clauses and contracts for data and code release.

data from NATS and ISAVIA database and ii) 132h of ATC speech data from different projects (including all the training data from Table 3.3), and we redirect the reader to [17] for further details. For now on, we refer to these datasets as *32h* and *132h* 'fine-tuning sets'. Later, we evaluate the low-resource scenario by fine-tuning E2E models with different amount of data, for this, we use NATS and ISAVIA as private databases, and LDC-ATCC and UWB-ATCC as public databases. Finally, we evaluate the WER shift by fine-tuning E2E models with audio data from different genders of ATCOSIM database.

Baseline hybrid-based ASR All experiments are conducted with Kaldi toolkit [100]. The baseline models are composed of six convolution layers and 15 factorized time-delay neural network ($\sim$31M trainable parameters). We follow the standard Kaldi's chain LF-MMI training recipe [153]. The input features are high-resolution MFCCs with online cepstral mean normalization. The features are extended with i-vectors. We use 3-gram ARPA LM during decoding. The model is trained for 5 epochs on 132h of ATC speech (that includes NATS and ISAVIA). This model is closed related to the one presented in Section 3.1. Further information and baseline performances can be found in our previous work [17, 19, 18]. SOTA WERs are listed in the last column of Table 3.3.

Table 3.3: Characteristics of public and private databases, from Table 2.2. The 32h train set includes NATS and ISAVIA, while the 132h set includes these and multiple datasets from Table 2.2. †baseline WERs with hybrid-based ASR trained on ATC data.

	Characteristics		
Dataset	Train / Test	SNR [dB]	WER [%]†
Private databases			
NATS	18h / 0.9h	≥20	7.7
ISAVIA	14h / 1h	15-20	12.5
LiveATC-Test	- / 1.8h	5-15	35.8
Public databases			
ATCO2-Test	- / 1.1h	10-15	24.7
LDC-ATCC	23h / 2.6h	10-15	-
UWB-ATCC	10.4h / 2.6h	≥20	-
ATCOSIM	8h / 2.4h	≥20	-

End-to-end ASR We use four configurations of Wav2Vec 2.0/XLS-R models. From now on, we refer to these models with the following tags: i) *w2v2-B:* BASE model;[3] ii) *w2v2-L:* LARGE-960h model;[4] iii) *w2v2-L-60K:* LARGE-960h-LV60K model;[5] iv) *w2v2-XLS-R:* XLS-R model.[6] We fetched all models' checkpoints from HuggingFace platform [155, 156]. Later, we perform standard fine-tuning with ATC speech data.

Hyperparameters end-to-end ASR We fine-tune each model for 10 k steps, with a 500-step warm-up phase ($\sim$5% of total updates). The feature extractor is frozen throughout the fine-

[3] 95M parameters, pretrained on train-set 960h LibriSpeech [94].

[4] 317M parameters pretrained and then fine-tuned with LibrSpeech 960h train-set.

[5] Same as w2v2-L but uses LibriSpeech + 60kh Libri-Light [154] during the pre-training phase.

[6] 300M parameters pretrained on 436kh of publicly available data in 128 languages [50].

tuning phase. The learning rate is increased linearly until $\gamma = 1\text{e}{-4}$ during warm-up, then it linearly decays. We use CTC loss function [45]. Dropout [157] is set to $dp = 0.1$ for the attention and hidden layers. We use GELU activation function [158] and AdamW [159] optimizer (β_1=0.9, β_2=0.999, ϵ=$1\text{e}{-8}$). We fine-tune each model on a single RTX 3090 with an effective batch size of 72. All the models use a character-based lexicon, i.e., we concatenate the English alphabet with symbols and the blank symbol, i.e., in total the vocabulary is composed of 32 characters. We use greedy decoding after applying Softmax to obtain the most likely character at each time step. Finally, we apply a data augmentation strategy similar to SpecAugment [160] as in previous work [49].[7] All E2E models in this section use the same set of hyperparameters.

Language modeling We concatenate all text transcripts and train 2/3/4-gram ARPA LMs. The LMs are integrated by shallow fusion with a Python CTC decoder, `PyCTCDecode`.[8] 4-gram LMs performed systematically better ($\sim$2% relative WER reduction) compared to 2-gram LMs in all test sets. We report results only with 4-gram LM, as in [49]. We set $\alpha = 0.5$ and $\beta = 1.5$, which corresponds to the LM and length normalization weights. We set the beam size to 100.

3.2.3 Incremental Training and Gender Bias

Incremental training With the recent success of SSL pretrained E2E models, it has become of particular interest to quantify how much data is actually needed to perform effectively on a downstream task. It is also important for low-resource tasks, such as ATC, where few tens of hours of labeled data are available for training or fine-tuning. In most ATC cases, data from one airport does not generalize well to other airports (for instance, see Table 3.5) due to a considerable AM domain-shift (accent, speaker rates and audio quality), as well as a LM domain-shift (dominance of different vocabulary). We analyze model performance versus different fine-tuning data sizes. We experimented with four *few-shot learning* scenarios with less than one hour ($\sim$1k utterances) of fine-tuning data. We split the experiments in two. First, we fine-tuned nine models on private databases, either NATS or ISAVIA data, as depicted on the left plot of Figure 3.2 (x-axis refers to number of utterances used during fine-tuning in log scale). Second, with the aim of open research, we performed the same approach on public databases, i.e., LDC-ATCC and UWB-ATCC. The results are on the right plot of Figure 3.2.

Gender experiments We use the free and open-source ATCOSIM database to carry the gender experiments. We obtained the gender labels for each utterance from the original ATCOSIM gold annotations.[9] We split the train set into increasing sizes of 1h, 2h, 3h, 3.5h, and also by gender. We aim at both, analyzing the performance in WERs caused by fine-tuning an E2E with audio from different gender, and to measure the performance gain by scaling up the fine-tuning data. We trained four models for each gender (using the same hyperparameters as the ones described

[7] We mask the input sequence with a probability $p = 0.075$, and $M = 12$ consecutive frames.

[8] Website URL: https://github.com/kensho-technologies/pyctcdecode

[9] Check our public GitHub repository for more details: https://github.com/idiap/w2v2-air-traffic.

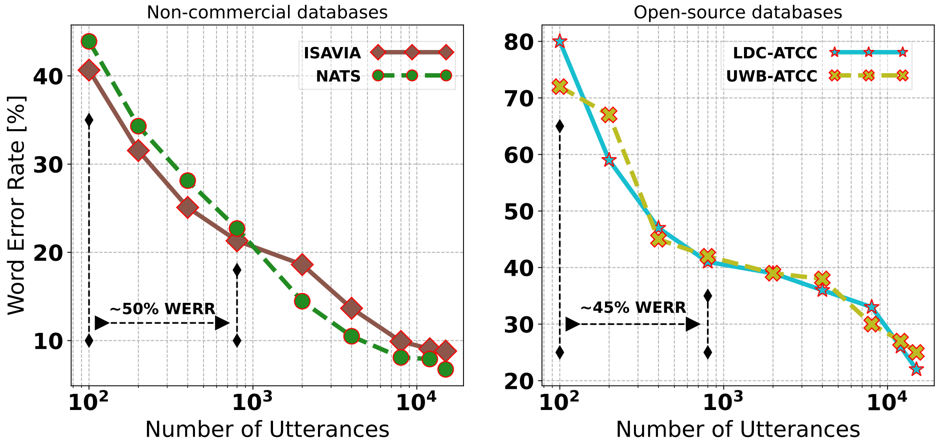


Figure 3.2: WERs for models fine-tuned with variable amount of utterances (x-axis) on both, private (left plot) and public (right plot) databases. Each data point corresponds to a train/test subset from the same dataset. 100, 1k and 10k utterances are roughly 5 min (few-shot), 1 h, and 10h, respectively. All the evaluations are reported with *w2v2-L-60K* model and without shallow fusion with in-domain LM. We also list the WER reduction (WERR) [%] by scaling up the fine-tuning set size from 100 to 800 utterances.

above and with the same model: *w2v2-L-60k*) and report the results in Table 3.6.

3.2.4 Results & Discussion

We structure the discussion of the results by addressing concrete questions. Our main hypothesis is that E2E models trained by SSL learn a robust representation of speech [49] and perform well on downstream tasks, i.e., ASR or multilingual ASR [50].

Breaking the paradigm, hybrid-based or E2E ASR? Although hybrid-based ASR modeling has been the default for several years, a new wave of E2E architectures pretrained by SSL for joint AM and LM is taking its place. We compare E2E models to our best hybrid-based ASR trained with the 132h fine-tuning set on Kaldi (**Baseline**, first row, Table 3.4). For E2E AMs we select two models. First, *w2v2-L-60k* to evaluate NATS and ISAVIA test sets, which is only fine-tuned on the 32h set, i.e., in-domain data. Second, *w2v2-XLS-R+* for ATCO2-Test and LiveATC-Test test sets, which is trained on 132h of ATC speech data [17, 19]. The 132h set is a more diverse set, and it was also used to train the hybrid-based baseline model. We obtained 30 and 41% relative WER reduction (WERR) on NATS and ISAVIA when using *w2v2-L-60k* instead of our hybrid-based ASR baseline. The improvement is considerable, even though the baseline model is trained on four times more data than *w2v2-L-60k* (see Table 3.4). Similarly, *w2v2-XLS-R+* (last row: Table 3.4) surpasses the hybrid-based model on all four test sets, but more significantly on the two most challenging, ATCO2-Test and LiveATC-Test sets. In total, 19 and 30% relative WERR on ATCO2-Test and LiveATC-Test were obtained, respectively

Table 3.4: WERs on four ATC test sets with with greedy decoding or beam search decoding with a 4-gram ARPA LM integrated by shallow fusion. Models are fine-tuned on NATS and ISAVIA. **Unlab. data column:** denotes the audio data using during the E2E pre-training stage: *LS*: LibriSpeech 960h train-set [94], *LV*: LibriVox 60kh train-set [154] and *ML*: 436kh of multilingual speech data [50]. *baseline WERs of Wav2Vec 2.0 [49] and XLS-R [50] models on LibriSpeech `test-other` set when fine-tuned on 10h of labeled data (comparable to our setup). †best Kaldi hybrid-based model (see [17, 19]) trained with the 132h set. ††models fine-tuned with 132h of ATC speech data (instead of 32h) and twice the number of steps, i.e., 20k. Numbers in **bold** refer to top WERs overall and underline with 132h set.

	Unlab.	**NATS**		**ISAVIA**		**ATCO2-Test**		**LiveATC-Test**		LS*
Model (θ)	**data**	Greedy	+LM	Greedy	+LM	Greedy	+LM	Greedy	+LM	-
Baseline (31M)										
Hybrid-based †	-	-	7.7	-	12.5	-	24.7	-	35.8	-
BASE (95M)										
w2v2-B	LS	10.7	8.4	12.5	10.1	45.6	40.1	48.1	42.2	7.8
LARGE (317M)										
w2v2-L	LS	9.3	7.6	11.7	9.5	44.9	40.0	47.5	41.4	6.1
w2v2-L-60k	LS+LV	**6.8**	**5.4**	**8.8**	**7.3**	34.6	31.2	39.8	34.5	4.9
w2v2-L-60k+††	LS+LV	9.3	7.4	11.2	9.1	23.3	21.2	31.1	27.2	-
XLS-R (300M)										
w2v2-XLS-R	ML	8.4	6.5	10.5	8.2	39.1	33.8	42.9	36.7	15.4
w2v2-XLS-R+††	ML	9.0	7.4	10.4	8.3	**22.8**	**19.8**	**29.7**	**24.9**	-

(hybrid-based → *w2v2-XLS-R+*).

However, it is worth mentioning that hybrid-based ASR is still considered the default implementation in many industrial applications due to some advantages over E2E models. Two examples are, hybrid-based ASR does not require high-performance computing (e.g., GPUs) to perform real-time inference, while E2E models relies heavily on GPUs for speed. Further, hybrid-based ASR can be easily deployed for streaming scenarios with minimum degradation on WERs. Yet, E2E models still involve considerable architectural modifications to attain comparable WERs [161, 162, 163].

Does additional partially in-domain data increases ASR performance? We answer this question by comparing models fine-tuned either on the 132h or 32h set. The former set is a mix of public and private databases, while the latter is only NATS + ISAVIA, thus private. Note that NATS and ISAVIA are clean in-domain ATC speech corpora, i.e., considered as in-domain on the 32h set and partially in-domain otherwise (132h set). Differently, ATCO2-Test and LiveATC-Test can be considered noisy and partially out-of-domain sets, i.e., airport, acoustic, and LM mismatch.

To address this question, we only focus on *w2v2-L-60k* and *w2v2-L-60k+* models fine-tuned on the 32h and 132h sets, respectively.[10] We analyze the WERs obtained by greedy decoding to focus only on joint acoustic and language ASR modeling (see Section 2.1.2). A degradation on WERs is observed for the in-domain test sets, NATS: 6.8% → 9.3% WER and ISAVIA: 8.8% → 11.2% WER. This is mainly to the addition of data that does not match NATS and ISAVIA. Contrary, there was considerable WER reduction on the partly out-of-domain sets, ATCO2-Test: 34.6% → 23.3% WER and LiveATC-Test 39.8% → 31.1% WER. NATS test set (ISAVIA: 1% relative WERR) was impacted by the addition of partly-in-domain data, i.e., ∼7% relative WER reduction. Nevertheless, challenging test sets improved dramatically, i.e., ATCO2-Test and LiveATC-Test 43% and 33% relative WERR.

Do multilingual pretrained E2E models help? To answer this question we compare *w2v2-L-60k+* and *w2v2-XLS-R+* models, which use the same hyperparameters, fine-tuning setup and beam search decoding with LM. We obtain a relative WERR of 8.8%, 6.6% and 8.5% on ISAVIA, ATCO2-Test and LiveATC-Test, respectively (no improvement on NATS). Significant improvement is seen on the most challenging test sets (SNR: 5-10 dB) which contain accented English speech, i.e., ATCO2-Test and LiveATC-Test. Hence, multilingual pretrained models bring a tiny, but noticeable boost in performance compared to single-language pretrained E2E models. This observed behavior can be attributed to the fact that *w2v2-XLS-R* have seen considerably more multilingual and accented audio data during the pre-training phase [50] in comparison to *w2v2-L-60k* [49].

The annotation process of ATC speech demands large amount of time. Thus, pre-transcription with an in-domain ASR model becomes an interesting path as it can subtantly decrease the overall annotation time. Following this idea, we believe that is of special interest for the research community to quantify how much audio data is needed to reach acceptable WERs for the ATC use case. We validated this idea by performing experiments with different amounts of fine-tuning data (utterances), thus it is up to the interested party to define the 'acceptable' WER threshold for the given application (e.g., deployment or pre-labeling only).

How much data do we need to fine-tune Wav2Vec 2.0 and XLS-R models? We also investigate the effect on WERs when different amounts of fine-tuning data are used during the fine-tuning phase. We divide the set of experiments by either using only public or private databases. The WERs on the private databases are given in the left plot of Figure 3.2. All the experiments are based on the most robust E2E model from Table 3.4 i.e., *w2v2-L-60K*.[11] The plots of Figure 3.2 denote the WERs for models evaluated with greedy decoding and without LM. We fine-tune 18 models varying the training data set (either NATS or ISAVIA) and varying the amount of fine-tuning samples. We initially tested the few-shot learning scenario ('worse-case'), where only 100 labeled utterances (∼5 min) were used for fine-tuning, and achieved WERs of

[10] Note that the results are still comparable for the XLS-R AM, i.e., *w2v2-XLS-R* versus *w2v2-XLS-R+*.

[11] We select the *best model* based on lowest WERs on out-of-domain test sets, i.e., ATCO2-Test and LiveATC-Test. See last row Table 3.4.

40% and 43.9% for ISAVIA and NATS. Further, ∼50% relative WERR is obtained by scaling up the fine-tuning data to 50 minutes (800 utterances). Specifically, NATS 43.9% → 22.7% WER and ISAVIA 40.6% → 21.3% WER. Lastly, if all available data (∼14h) is used, we reach an 8.8% and 6.8% WER for ISAVIA and NATS, respectively. This represents an ∼80% relative WERR compared to the low-resource setup (100 utterances). With around 8h (∼8000 utterances), *w2v2-L-60K* beats the performance of our SOTA hybrid-based ASR (which uses four times more training data). We follow the same methodology to evaluate the public databases. We also train 9 models for each dataset, i.e., LDC-ATCC and UWB-ATCC. We list the WERs on the right plot of Figure 3.2 for both test sets. Here, we note similar behaviors, thus we reach similar conclusions. First, scaling-up the fine-tuning data from 5 to 50 minutes brought ∼45% relative WERR for both, LDC-ATCC and UWB-ATCC test sets (similar trend in private databases, NATS and ISAVIA). Not surprisingly, further gains in WERs are achieved if we increase the fine-tuning data up to 11h. Previous research has not explored E2E modeling[12] in the area of ATC, thus, these WERs can be adopted as baselines.

Transferability between ATC corpora We have stated before that E2E models fine-tuned on a specific ATC corpus might not transfer well to different ATC corpora.[13] To test this hypothesis, we train models with different public databases and test them on four test sets which reflect different ATC scenarios (e.g., data not seen during training). We fixed the model (*w2v2-L-60k*), training data size to 11h, and same hyperparameters. From Table 3.5, we can conclude that UWB-ATCC corpus transfers better to different databases, for instance LDC-ATCC. In this case, if we fine-tune *w2v2-L-60k* with UWB-ATCC set and test it on LDC-ATCC the performance is 54% WER, whereas inversely the performance is 64%, i.e., ∼10% absolute WERR. Similarly, the model trained on UWB-ATCC fits better ATCO2 test by a large margin compared to LDC-ATCC, i.e., 10% absolute WER reduction.[14]

Table 3.5: WERs on different test sets. Models are fine-tuned only on public databases and fixed to 11h of audio data. All systems are *w2v2-L-60k* and WERs are obtained with greedy decoding and no LM. †test set split by gender (Male/Female).

	Test set			
Train set	LDC	UWB	ATCO2	ATCOSIM (M/F)†
LDC-ATCC	**25.0**	64.1	58.7	41.1 / 35.7
UWB-ATCC	54.6	**21.9**	**47.9**	**32.5 / 24.6**

Gender bias on ATC speech We analyze the gender bias on ATCOSIM dataset, which provides the gender labels for each utterance. The results are listed in Table 3.6. It is evident that the experiments with female voice performed systematically better in all training scenarios (1h to 3.5h fine-tuning set). We also aimed to test the possibility that the speech rate was the main cause of this behavior. In average, each female recording has a speech rate of 3.4 words per second

[12] Training scripts to replicate the right plot of Figure 3.2 are public in our GitHub repository.

[13] This assumption also applies to hybrid-based ASR models.

[14] This conclusion can be supported because UWB-ATCC training data partially matches ATCO2 test set signal quality.

Table 3.6: WERs on ATCOSIM for models fine-tuned with *w2v2-L-60k* and greedy decoded. We experiment with different fine-tuning set sizes. WERs are reported on 0.7h of speech (only from the same gender) sampled from the original test set, i.e., train-test within the same gender. †list the WER reduction by scaling from 1h→3.5h.

	Dataset size				**WERR**†
Gender	1h	2h	3h	3.5h	(1h →3.5h)
Male	36.70	31.42	29.20	**28.72**	21.74%
Female	17.62	13.91	13.46	**12.37**	29.79 %

(WPS) while male has an average of 2.9 WPS.[15] In order to determine whether female recordings are of better quality than the male ones, or whether the E2E model have some bias acquired during the pre-training phase, we calculated the WERR when fine-tuning the model between 1h to 3.5h of audio. Following Table 3.6 we can see that in the female experiments the reduction on WERs is higher than on the male side by around 8% absolute when scaling from 1h to 3.5h.

We believe that E2E models (e.g., Wav2Vec 2.0) might carry little but noticeable gender bias. For instance, previous work have concluded that gender unbalance might affect E2E models during the pre-training phase [151]. However, this bias can be mitigated by adding a small amount of data from the opposite gender [150]. In conclusion, it is still prudent to perform more thorough experiments before reaching hard judgments in this regard, or at least, in ATC communications.

Conclusions Our experiments show large recognition improvements of Wav2Vec 2.0 and XLS-R compared to *hybrid-based* ASR baselines. Quantitatively, between 20% and 40% relative WERR was obtained on ISAVIA and NATS test sets, but also on challenging databases with multiple accents, i.e., ATCO2-Test and LiveATC-Test. Furthermore, we demonstrated that pretrained models allow rapid fine-tuning with small quantities of adaptation data. Finally, this is the first research aiming at analyzing the performance of large-scale SSL acoustic models on ATC.

[15] WPS computed from the training sets.

3.3 Using Contextual Knowledge for Hybrid-Based ASR

Automatic speech recognition (ASR), as the assistance of speech communication between pilots and air-traffic controllers, can significantly reduce the complexity of the task and increase the reliability of transmitted information. ASR engines can lead to a lower number of incidents caused by misunderstanding and improve air traffic management (ATM) efficiency. Evidently, high accuracy predictions, especially, of key information, i.e., callsigns and commands, are required to minimize the risk of errors. We prove that combining the benefits of ASR and NLP methods to make use of surveillance data (i.e., an additional modality) helps to considerably improve the recognition of key entities (named entities) in the ATC speech, i.e., callsigns.

In this work, we investigate a two-step key entities boosting approach:

- (1) ASR step: weights of probable callsign n-grams are boosted in G.fst and/or in the decoding FST (lattices);
- (2) NLP step: callsigns (named entities) extracted from the improved ASR transcript with NER are correlated with the surveillance data to select the most suitable one.

Publication Note

The material presented in this section is adapted from the following publications:

- I. Nigmatulina, J. Zuluaga-Gomez, A. Prasad, S. S. Sarfjoo, and P. Motlicek, “A two-step approach to leverage contextual data: speech recognition in air-traffic communications,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6282–6286
- M. Kocour, K. Veselý, A. Blatt, J. Zuluaga-Gomez, I. Szöke, J. Černocký, D. Klakow, and P. Motlicek, “Boosting of Contextual Information in ASR for Air-Traffic Call-Sign Recognition,” in *Proc. Interspeech*, 2021, pp. 3301–3305
- I. Nigmatulina, S. Madikeri, E. Villatoro-Tello, P. Motlicek, J. Zuluaga-Gomez, K. Pandia, and A. Ganapathiraju, “Implementing Contextual Biasing in GPU Decoder for Online ASR,” in *Proc. Interspeech*, 2023, pp. 4494–4498

Supplementary materials related to this section:

- **Code - GitGub repository at:** https://github.com/idiap/contextual-biasing-on-gpus
- ATCO2 project website: https://www.atco2.org/

Major contributions Problem definition and experimental design and setup. Data preparation. Proposed the idea of integrating contextual information with a second NLP step, leading to substantial improvements in callsign recognition accuracy. Trained the hybrid-based ASR systems. Actively participated in the article write up.

3.3.1 Introduction

There are multiple key entities in speech communication between pilots and Air-Traffic Controllers (ATCo), i.e., callsigns, which are used for identification of aircraft. ASR systems aiming to recognize callsigns in real time while requiring high recognition accuracy (below $\sim$5% WER). Particularly, the callsigns are unique aircraft identifiers, of which the first part is an abbreviation of the airline name and the last part is a flight number that contains a digit combination and may also incorporate an additional character combination, e.g., *TVS84J* (see Table 3.7). At a certain time point, only few aircraft are usually in the radar zone, which means only a limited number of callsigns can be referred to in the ATCo communications. If a recognized callsign does not match any 'active' callsign registered by radar at the given time point, it means that there is no corresponding aircraft in the airspace and the automatically recognized command (from voice communication) is invalid. Therefore, contextual information coming from the surveillance (radar) data allows adjusting ASR system predictions that can significantly increase its accuracy. However, the use case of combining contextual knowledge with input speech does not mean the ASR scenario can be considered as closed-set callsign recognition. Instead, the solution is implemented with an ASR and open-set list of callsigns, since in scenarios such as "en-route" different aircraft can appear on the communication frequency listened by ATCo.

Table 3.7: Callsigns: compressed and extended (airlines designators are in bold)

Callsign	Extended callsign
SWR2689	**swiss** two six eight nine
RYR1RK	**ryanair** one romeo kilo
RYR1SG	**ryanair** one sierra golf

Although contextual information has been already used in previous ATC studies for ASR [164, 165, 166, 167], or more recently in [21, 26, 17]; it has never been adapted for both, ASR and concept extraction outputs simultaneously and without a need of any additional knowledge (e.g., manual annotation, well-defined classes, etc.). This research aims to leverage the available contextual information by combining ASR and NLP methods. We believe that ASR and NLP are complementary tasks rather than standalone ones. Whereas ASR exploits speech to produce a sequence of words, NLP exploits the intrinsic characteristics of text for a downstream task. ASR normally struggles to model long sequences, while state-of-the-art NLP systems allow extracting key information from whole chunks of text; for instance, an entire ATC utterance. In the proposed approach, we focus on an iterative use of contextual data to take advantage of a combination of ASR and NLP modules. (1) First, boosting the probability of active callsigns in ASR system (*FST-boosting*), (2) second, boosting ASR outputs (*NLP-boosting*) in order to correct those predicted callsigns, which are not present in the surveillance data.

3.3.2 Contextual Biasing for Hybrid-based ASR

Contextual data within ASR systems can be integrated by modifying weights of target n-grams in the grammar or/and in the ASR output lattices, e.g., by mean of generalized composition of an in-domain LM and Weighted Finite State Transducers (WFSTs) with the target contextual n-grams [168, 169, 170]. A similar approach has been recently adopted in the ATC domain [22, 21, 29] and proved to offer a significant gain in callsign recognition. A list of callsigns to be boosted is regularly changing and needs to be updated dynamically per each utterance. Thus, weights of callsign n-grams are dynamically modified in the WFST. The first of the methods is **lattice rescoring**, where the weights are adjusted on the word recognition lattices from the first pass decoding. In the other method, weights are dynamically modified directly in the grammar (**G.fst**), which allows having target n-grams boosted before the decoding is performed [29]. For our experiments, we will adopt the lattice rescoring approach. Besides aiming to reduce WERs, contextual information for ATC has been also used to improve concept extraction [164, 165, 166, 167]. Schmidt et al. [164] applied a Context-Free Grammar (CFG)-based LM, limiting the search space according to the contextual data. Shore et al. [165] and Oualil et al. [166, 167] build a CFG-based concept extractor with all semantic concepts of ATC embedded in XML annotation tags.

3.3.3 Step 1–Injecting Contextual Knowledge During ASR Decoding

In hybrid-based ASR systems, the different knowledge sources are represented as WFSTs, which are combined by the 'composition' operator together in the final decoding graph [171]. Information from additional knowledge sources can also be integrated into a system by means of composition. Our first integration of contextual knowledge into ASR is done on the LM level (*G-extension*). The idea is to boost callsign n-grams already available in LM, and even more important to add those callsign n-grams, which are absent (e.g., $>$3 words sequences in 3-gram LM). We build a contextual *FST* that includes all possible callsigns per utterance: all callsigns registered by the radar at different time stamps (from 17K to 280K callsigns to boost in different test sets; see last column in Table 3.8). Then, the main $G.fst$ is composed with the contextual $G\_biased.fst$ and the result of composition is used in the final decoding $HCLG$ graph. The second integration of contextual information (*lattice rescoring*) is done per utterance on top of the decoding lattices which allows flexible adaptation to new-coming contextual information avoiding changing the main decoding graph ($HCLG$) (see [29]). Weights in lattices are rescored according to the surveillance data: for each test utterance, a FST biased to callsigns n-grams registered at the time stamp when an utterance is created and composed with lattices created in the first pass:

Table 3.8: Test sets with callsigns per utterance (csgn per utt.) — median of callsign per utterance in the surveillance data.

Test set	N of utt with	N of utt w/o a csgn	Csgn per utt	Min	All csgns
LiveATC	581	29	28	40	280K
M-Prague	784	88	5	82	17K
M-Vienna	877	38	19	65	59K
NATS	794	73	50	50	168K

$$Lattices' = Lattices \circ biasing\_FST, \tag{3.1}$$

weights updated in the composition are used for final predictions.

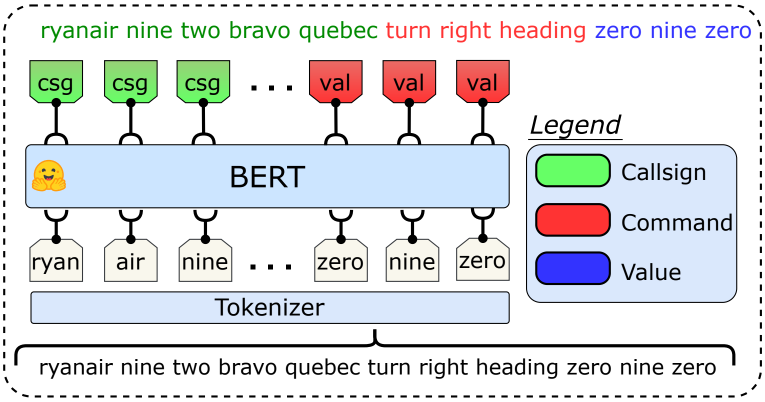

ryanair nine two bravo quebec turn right heading zero nine zero
csg
csg
csg
. . .
val
val
val
Legend
BERT
Callsign
Command
Value
ryan
air
nine
. . .
zero
nine
zero
Tokenizer
ryanair nine two bravo quebec turn right heading zero nine zero

Figure 3.3: BERT-based NER pipeline.

3.3.4 Step 2–Injecting Contextual Knowledge Post-ASR Decoding

Our approach for integrating contextual knowledge on ASR transcripts (e.g., 1-best hypothesis) is based on a two-step pipeline. Each step conveys an independent module, as below.

Step 2.1–Named Entity Recognition (NER) module ATC communications carry rich information such as callsigns, commands, values, and units; they can be seen as 'named entities'. We propose a NLP-based system to extract such information from ASR transcripts. We defined callsigns, commands, units, values, greetings, OR the rest (e.g., 'None' class) as tags for the NER task, as depicted in Figure 3.3. First, we download a BERT [87] model from Huggingface [155] and then fine-tune it on NER task. We use 12k sentences (∼12h of speech), where each word has a tag. Specifically, we use the LiveATC database for this purpose, see more information in [18]. We developed a data augmentation pipeline in order to increase the amount of training data for NER, i.e., we generated 1M samples from the initial 12k sentences. The pipeline has four actions that modify the training sample: *add*, *delete*, *swap*, or *move* the **callsign** across the utterance–sentence–. *Delete* and *move* actions, remove and keep the same callsigns, respectively; *add* and *swap* generate a sentence with a new callsign picked randomly from a pre-defined callsign list by the user. This makes the approach flexible and easy to deploy in multiple settings.

Step 2.2–Re-ranking module based on Levenshtein distance The BERT-based system for NER allows us to extract the callsign from a given transcript or ASR 1-best hypotheses. Recognition of this entity is crucial where a single error produced by the ASR system affects the whole entity (normally composed of three to eight words). Additionally, speakers regularly shorten callsigns in the conversation making it impossible for an ASR system to generate the full entity (e.g., *'three nine two papa'* instead of *'austrian three nine two papa'*, *'six lima yankee'* instead of *'hansa six lima yankee'*). One way to overcome this issue is to re-rank entities extracted by the BERT-based NER system with the surveillance data. The output is a list of tags that match words or sequences of words in an input utterance. As our only available source of contextual knowledge are callsigns registered at a certain time and location, we extract callsigns with the NER system and discard other entities. Correspondingly, each utterance has a list of callsigns expanded into word sequences (shown in Table 3.7). As input, the re-ranking module takes (i) a callsign extracted by the NER system and (ii) an expanded list of callsigns. It then compares a given n-gram sequence against a list of possible n-grams, and finds the closest match from the list of surveillance data based on the weighted Levenshtein distance. We skip the re-ranking in case the NER system outputs a 'NO_CALLSIGN' flag (no callsign recognized).

ASR modeling The acoustic model is a CNN-TDNNF trained on approximately 1200 hours of ATC labeled augmented data [18, 19] with the Kaldi framework [100]. First, the training databases (195 hours) were augmented by adding noises that match LiveATC audio [18] channel (one batch between 5-10 dB and other 10-20 dB SNR). Afterward, we applied speed perturbation, i.e., generates 1200 hours of training data. The model was further improved with 700h of semi-supervised data collected in LiveATC for different airports from Europe [139].[16] The LM is 3-gram trained on the same data as the AM with an additional corpus from various public resources such as airlines names, airports, ICAO alphabet and way-points in Europe.

3.3.5 Results & Discussion

As a baseline, we use callsign extraction done directly on the outputs of our ASR system. Then, we apply the proposed boosting techniques (G-extension, lattice rescoring, NLP-boosting) in different combinations to see how they can benefit from each other. In Table 3.9, the results of the experiments are presented on four different test sets with accuracy of callsign (ICAO) recognition. Overall, the proposed metrics help to improve the baseline accuracy from 30.6% to 53.7% absolutely, or from 32.1% to 60.4% relatively (for the test sets Prague and NATS correspondingly; when the NATS set gets the highest improvement being the out-of-domain data). The best results are always achieved with the use of NLP-boosting. For LiveATC and NATS sets, the out-of-domain sets in the ASR training, the best performance is achieved with the combination of NLP-boosting and ASR-boosting (lattice rescoring) methods.

At the same time, the G-extension has a contradicting effect. It helps to improve results comparing to the baseline for the LiveATC and Vienna sets, yet, its combination with lattice rescoring achieves worse accuracy than lattice rescoring alone. The possible drawback of the G-extension method is that a very high number of available callsigns are boosted in LM FST (see last column 3.8). It can introduce confusion when combining with the lattice rescoring boosting method, which focuses on only current callsigns. Also, this does not need any modifications during the decoding and serves as a general domain adaptation. Thus, G-extension can be used to improve the outputs when other methods are not available, otherwise, we skip it. The number of callsigns used to boost the ASR outputs may also degrade the performance of lattice rescoring.

[16] Note that the hybrid-based ASR system in this work is more robust w.r.t the one presented in Section 3.1 due to the data augmentation process.

Table 3.9: Results of callsign extraction with ASR boosting (ASR-B) and post-boosting (NLP-B): the accuracy of callsign recognition (%) is calculated for the callsigns in ICAO format.

Method			**Test sets (callsign recognition accuracy)**			
			LiveATC	**Prague**	**Vienna**	**NATS**
ASR outputs + *Callsign extraction* (baseline)			42.8	64.4	48.4	35.2
Lattice rescoring	**G-extension**	**NLP-boosting**				
✓	-	-	53.1	66.9	59.6	37.1
-	✓	-	44.4	64.3	49.2	34.8
✓	✓	-	52.8	66.9	52.1	36.8
-	-	✓	88.4	**95.0**	**86.0**	87.0
✓	-	✓	**88.5**	94.8	84.3	**88.9**
-	✓	✓	87.7	**95.0**	85.6	88.2
✓	✓	✓	88.0	94.7	84.0	88.0
Gold annotations + *Callsign extraction* (oracle)			**89.7**	72.2	59.6	67.4
+ NLP-Boosting			89.3	**95.4**	**87.0**	**94.0**
ASR WER (without boosting)			32.4	3.4	9.2	24.4

Although in this case, the number of callsigns did not exceed 50, we investigated its impact. The test sets have different numbers of boosted n-grams, from 5 to 50 (see Table 3.7), but even with 50 boosted callsigns the recognition accuracy goes considerably up comparing to the baseline. Along with the evaluation of boosting methods on the ASR outputs, we provide the ‘oracle’ results, when callsigns are extracted on the ground truth transcriptions (**Gold annotations** line in Table 3.9). This comparison allows estimating the impact of the proposed methods to the callsign extraction improvement, when no ground truth information is available. Even if the ‘oracle’ scores always stay better, the accuracy achieved with our systems shows close and comparable results. No improvement with NLP-boosting on the ground truth transcription for LiveATC test set can be explained by already high accuracy of callsign extraction, as LiveATC data was used to fine-tune the NER. Our methods demonstrate consistent results for data of different quality. The level of noise in the recordings of LiveATC and MALORCA test sets is very different, as well as WERs achieved by their baseline ASR systems (the last line in Table 3.9; [29]). Nevertheless, we see considerable improvement for all test sets and the general tendency stays the same.

Conclusion We investigated a two-step approach of integrating contextual radar data in order to dynamically improve the recognition of callsigns per utterance. We demonstrated that the best result is achieved with (1) NLP-boosting and (2) NLP-boosting+lattice rescoring methods on all test sets of different recording quality with the significant improvement, i.e., from 32.1% to 60.4% of relative improvement on callsign recognition accuracy across the evaluated data sets. Introduction of contextual information considerably improves recognition of callsigns and, thus, recognition of ATCo messages in general. As a noisy environment leading to lower recognition accuracy is often a reality in pilot-ATCo communication, the proposed methods and their combination will definitely benefit the recognition of the key information in ATCo speech.

3.4 Using Contextual Knowledge for End-to-End ASR

Despite the recent success of end-to-end models for automatic speech recognition (ASR), recognizing out-of-vocabulary (OOV) words, rare words, and fast domain adaptation with text, are still challenging. We propose a light *on-the-fly* method to improve ASR performance with the shallow fusion of an n-gram language model (LM) with the Aho-Corasick (AC) string matching algorithm. The AC algorithm has proved to be more efficient than other methods and allows fast context adaptation. An n-gram LM is introduced as a graph with fail and output arcs, where the arc weights are adapted from the n-gram probabilities. In addition, our method is used as a support to keyword biasing when the LM is combined with the context bias entities to improve the overall performance. We demonstrate our findings on 4 languages, 2 public and 1 private datasets, including the performance on named entities and OOV words.

Our contributions are covered below:

- To the best of our knowledge, this is the first use of the AC algorithm to integrate word-level n-gram LMs, previously used only for keywords biasing;
- combining n-gram LM with keyword biasing in a single trie;
- as English is the dominant language in SF studies, we extend our evaluation to 3 other languages from CommonVoice;
- analysis of the method performance on NEs, OOV words, and real-time factor (RTFX) measures.

Publication Note

The material presented in this section is adapted from the following publication:

- I. Nigmatulina, J. Zuluaga-Gomez, *et al.*, "Improved contextual adaptation with an external n-gram language model for Transducer-based ASR," in *Submitted to INTERSPEECH 2024*, 2024

Supplementary materials related to this section:

- **Code - GitGub repository at:** https://github.com/idiap/contextual-biasing

Minor contributions Problem definition and experimental design and setup. Trained the Transformer-Transducer models for experiments on CommonVoice. Data curation and creation of the biasing lists using a pre-trained BERT model on the NER task. Trained the Transformer LM for the experiments. Actively participated in the article write up.

3.4.1 Introduction

Available contextual text data (in-domain text data, specific terminology, proper names, etc.) can considerably improve the performance of ASR. With the recent advance of End-to-End (E2E) speech recognition and its replacement of the hybrid models, the dynamic incorporation of contextual information and text-domain adaptation of the E2E models is still an open research question. It is mainly due to the difficulty of adapting the internal language model (ILM; [172, 173, 174]) which is implicitly trained with the overall E2E loss, compared to the stand-alone language model (LM). It is also a reason why many decoding methods aim to integrate an external LM [175]. The traditional methods of integrating contextual text data are divided into two directions: (1) when the information is introduced during the decoding without any change in the ASR architecture, i.e., *rescoring* and *shallow fusion* (SF) [176, 175, 177, 178], and (2) when the model is trained to be able to accept context when needed [179, 180]. In the latter group of methods, the ASR model usually includes an additional contextual module and needs the corresponding train data. Since it is not always possible to train a separate specialized model, we focus on the first group of methods, which can be applied to any ASR model and are considered the least costly methods of context integration, where the context is integrated directly during decoding with beam search.

Shallow fusion means log-linear interpolation of the score from the E2E model with an external contextual LM. SF of an external LM typically yields WER reduction in the target domain. Doing SF with a list of target entities [69, 177] can bias the hypotheses towards particular words or named entities (NE), e.g., proper names, terminology, geographical names, etc. For the speed and convenience of the decoding, contextual information is usually presented as a graph: n-gram LM – as a Weighted Finite-State Transducer (WFST) [171, 26], bias list – as a prefix Trie [177].[17] The disadvantage of the standard WFST and keyword prefix trie algorithm is that they do not support the mismatch cases in a search trie when the search string fails.

Recent studies show the efficiency of the classical string-matching Aho-Corasick (AC) algorithm [181] for NLP tasks [182] and keyword biasing in ASR [178]. Inspired by these works, we apply the AC algorithm for SF with a simple n-gram LM in ASR beam search decoding. Since [178] uses AC to improve the performance of keyword biasing, additionally, we propose an extension of the method by combining keywords with the LM n-grams and building a unified context graph (see Figure 3.4). We demonstrate our results with a Transformer-Transducer model (Zipformer-based encoder [58]) in four languages and three datasets.

Related work SF can be seen as a dynamic rescoring strategy that happens during beam search decoding and before pruning [176]. SF refers to log-linear interpolation between the ASR outputs and a separately optimized language model (LM) at each step of the beam search:

$$y^* = \arg\max \log P(y|x) + \lambda \log P_C(y), \tag{3.2}$$

where $P_C(y)$ is an in-domain or context-biased *contextual* LM and λ is a hyperparameter to control the impact of the contextual LM on the overall model score [176, 175]. Authors in [176] show the effectiveness of SF with neural-network LM (NN-LM) at reducing error compared to the n-gram LM. SF with NN-LM, however, considerably slows down the decoding and is not suitable for fast context change. Moreover, training an NN-LM demands a lot more data than training a statistical n-gram LM which can be an obstacle in the low-resource scenario. Furthermore, [175] improve SF with biasing at the subword unit level instead of word level and

[17] https://github.com/kensho-technologies/pyctcdecode

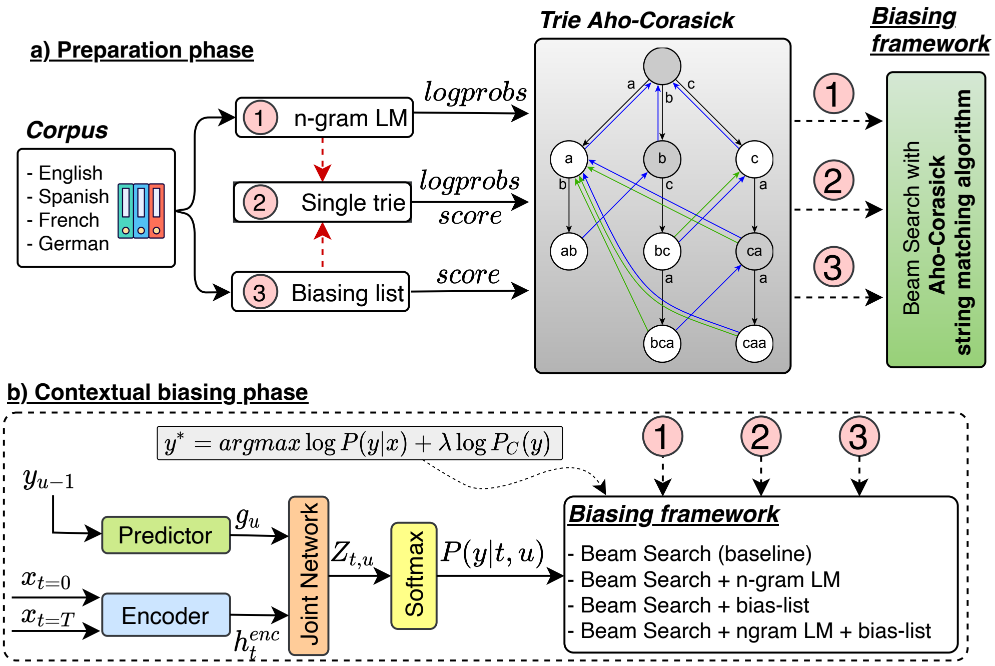


Figure 3.4: Proposed biasing approaches at beam search time with Aho-Corasick string matching algorithm. This approach works with Transformer-Transducer models with negligible speed reduction w.r.t beam search alone.

use a set of common prefixes (*"call"*, *"text"*) to avoid adding irrelevant bias. [177] integrate context keywords directly with the *keyword prefix trie* search algorithm without a need to train any LM. Our work is the most similar to [178], where they improve the keyword prefix trie method for context biasing with the AC algorithm. The main difference is that we apply the AC algorithm with the word-level n-gram LM and further combine it with keyword biasing.

3.4.2 Contextual Biasing with Aho-Corasick Algorithm

Aho-Corasick algorithm The AC algorithm proposed by [181] is a text pattern string matching algorithm where the search is done in linear time. The algorithm has three data structures as a representation of a search set, or a *transition diagram*: a trie, an output table, and a failure function (Figure 3.5). The output table is a set of suffixes reached from any node which are the target search strings. The failure function is applied when some search strings may be suffixes of others [183]. For example, if a trie includes the word "CAN" but does not include the word "CAT", when the string "CAT" fails to match, the failure transition will backtrack it to the prefix **CA–** from "CAN". The algorithm is implemented by building a finite state machine that allows backoff arcs not only to get the root of the graph or the lower-order n-gram (e.g., from a 4-gram to a 3-gram) when a string end is reached but also to do *failure transitions* to backtrack a string in case of its fail. This allows finding partial matches and also makes the arrays of the trie sparse, which helps improve the efficiency of the algorithm since the prefix match is not duplicated.

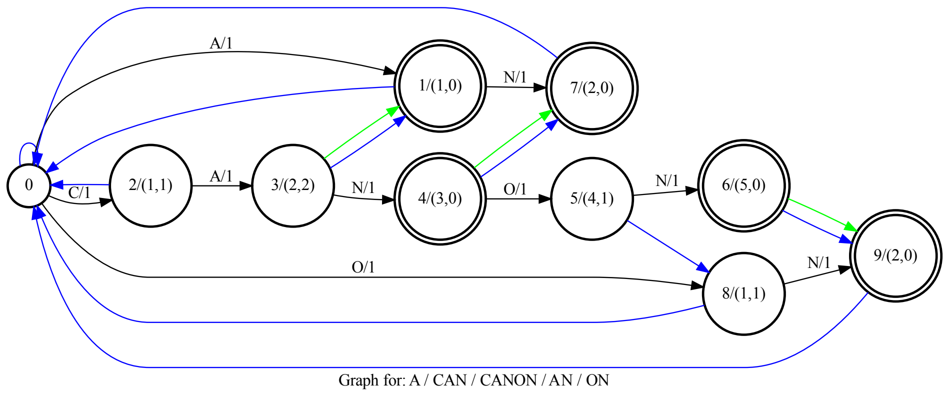


Figure 3.5: Aho-Corasick trie: blue lines are fail arcs and green lines are output arcs.

Shallow fusion with the n-gram word level LM The main idea of our approach is to apply the AC algorithm to SF with word-based n-gram LM, and thus, build an AC prefix trie directly with LM n-grams and LM log probability weights. The Transducer model we use is trained at the BPE units level [67]. However, the n-gram LM we integrate is word-based, as it provides word-level statistics and greater transparency in case of modification or biasing. Since the ASR model outputs its hypotheses at the BPE level, the LM n-grams are first converted into strings of BPE units with SentecePieces [68].[18] During keyword biasing in decoding, whenever a string match occurs between a hypothesis and a string in the context trie, a fixed positive cost is added to the log probability of the matched candidate. In the case of n-gram LM, we want to use LM weights to ensure balanced bias across all LM n-grams.

Assuming n-gram LM weights (e.g., in ARPA) are on a logarithmic scale, i.e., log probabilities, and aiming to find some positive costs that would best correspond to these LM weights, we convert LM log probabilities back to probabilities by taking an exponent. Although the n-gram weights are log-based 10 probabilities, experimentally applying exponential with base e instead of 10 shows better adaptation results on our datasets, and all further reported results are achieved with exponential based-e.[19] For each n-gram, a cost received from its word-level LM weight is assigned to each subword arc of the n-gram. Not dividing the n-gram cost by the number of subwords pieces gives the best improvement.

To control for the influence of word-level statistics integrated on the subword level, we evaluated the performance of OOV words. Moreover, the AC algorithm's ability to find partial matches is beneficial for OOV words recognition.

[18] https://github.com/google/sentencepiece

[19] We followed this approach based on experimental results. A larger hyper-parameter search or a scaling parameter could help to avoid this step.

Contextual biasing with the n-gram LM An AC-based trie is proposed to bias keywords in [178] and a natural extension of the described method is to combine an n-gram LM with target keywords. To combine LM n-grams and keywords, a single context trie is built. First, all the n-grams of the LM are added with their respective weights, and then the keywords are added with a bias cost depending on whether the keyword n-gram is present in the LM and already has some weight, or is not in the LM. The costs for *in-LM* and *out-of-LM* keywords were tuned for different datasets, and for most of our datasets they are 1.0 and 1.5 respectively.

Implementation For our experiments, we use the *k2/Icefall* framework that provides training recipes and decoding scripts including an implementation of the AC algorithm for keyword biasing.[20] Decoding uses subword-level beam search, which we adapted for SF with the AC trie and a word n-gram LM. For all the experiments, we use the same Zipformer-Transducer model, and the difference in results is only due to decoding. The baseline is the default beam search without SF. The other experiments include SF with the following external contexts: (1) Transformer-LM, (2) n-gram LM, (3) n-gram LM with the AC-based trie, (4) keyword biasing with the AC-based trie, (5) the combination of n-gram LM and keyword biasing with the AC-based trie. As fast and flexible context integration is critical in many practical scenarios, we include an estimate of decoding time using the *inverse real-time factor (RTFX)*, which is the ratio between the length of the processed audio and the decoding time.

3.4.3 Experimental Setup

Dataset description As our experiments include keyword biasing, besides audio and corresponding transcriptions we also need keyword lists with entities to bias. There are only a few publicly available test sets that satisfy this criterion, mostly in English. We evaluate the proposed biasing approaches on one private (*DefinedAI*.[21] banking, insurance, and healthcare domain) and two public datasets (*Earnings21*,[22] stock market domain [185]; and *CommonVoice* [97]); see Table 3.10. The DefinedAI and Earnings21 datasets have gold named-entities (NEs) that we use as bias lists: DefinedAI has manually annotated NE tags within each transcription (the main reason why we use this data set), Earnings21 has two general biasing lists based on the NER[23] [184]. The

Table 3.10: Test sets with context information (statistics). †utterances with at least one NE.

Test set	Size	Duration	Biasing entities	
		(hours)	unique	nb. utt†
DefinedAI	2K utt.	6	367	486
Earnings21	18K utt.	39	1013	-
CV-EN	16K utt.	27	1173	1125
CV-DE	16K utt.	27	1985	1906
CV-FR	16K utt.	26	600	549
CV-ES	15.5K utt.	26	122	135

[20] https://github.com/k2-fsa/icefall/blob/master/icefall/context_graph.py

[21] Private dataset obtained from an internal industrial project. See more information on DefinedAI website: https://www.defined.ai

[22] We split audios into 3-minute segments for decoding, as in [184].

[23] The *oracle* and the *distractor* lists are released by [184]. We employ the *oracle* list only.

CommonVoice public dataset is chosen to provide evidence on different languages, but it does not have gold annotated entities, so we prepared the NE bias lists ourselves.

Biasing lists for CommonVoice To create bias lists for CommonVoice, we use BERT models fine-tuned on the NER task to label, extract, and collect NEs for English (EN), German (DE), French (FR), and Spanish (ES) test sets.[24] We download the checkpoints from HuggingFace [155] and run NER for each language individually. We then remove all the single-word NEs, e.g., unigrams, to reduce noisy outputs that can hinder the biasing approach. The statistics after running this approach are in Table 3.10. Experiments on CommonVoice serve two purposes: (1) evaluate the SF approach on non-English languages and (2) see the impact of biasing lists with numerous unique entities, e.g., DE subset has ∼2k unique entities. Note that DE and EN languages contain more unique entities w.r.t FR and ES.

3.4.4 Model Training & Evaluation

Transformer-Transducer Training For all the experiments, we use Zipformer stateless transducer model [55] proposed and described in detail in [58]. For evaluation on DefinedAI and Earnings21 test sets, we take the pretrained Zipformer model on Gigaspeech-XL dataset [95].[25] as Earnings21 has only test data and for DefinedAI we have access to only a small amount of train data, i.e., 50h. The choice of training data is motivated by a previous study on Earnings21, where the authors use Gigaspeech as training data [184], and we can consider this scenario close to domain adaptation, as DefinedAI and Earnings21 are domain-specific data.

For experiments on CommonVoice, we train Zipformer models for each language on the corresponding train set; we train from scratch with the latest Icefall Transducer recipe and its default training hyper-parameters. This includes *ScaledAdam* optimizer [186] and learning rate scheduler with a 500-step warmup phase [61] followed by a decay phase dictated by the number of steps (7.5k) and epochs (3.5 epochs) [58]. The neural Transducer model is jointly optimized with an interpolation of simple and pruned RNN-T loss [60, 53] and CTC loss [45] ($\lambda = 0.1$). The learning rate peak is set to $lr = 5.0e^{-2}$ and we train each model for 30 epochs on a single RTX 3090 GPU.

Language modeling For shallow fusion with AC algorithm, we train 3-gram word-level LMs with SRILM [147]; for SF without AC algorithm, we train Transformer-based [61] LMs and 5-gram BPE LMs. To train n-gram LMs, for all test sets except Earnings21, we use text data from the corresponding train sets. For Earnings21, we use transcriptions from Earnings22 [187], which is a different dataset but from the same domain. To train Transformer LMs, we use GigaSpeech-XL text data for DefinedAI and Earnings21 and language-specific CommonVoice

[24] ES: mrm8488/bert-spanish-cased-finetuned-ner; EN: dslim/bert-base-NER-uncased; FR: cmarkea/distilcamembert-base-ner; DE: fhswf/bert_de_ner.

[25] Gigaspeech-XL: 10kh of transcribed audio data, model: yfyeung/icefall-asr-gigaspeech-zipformer-2023-10-17

Table 3.11: SF for the out-of-domain evaluation with Zipformer Giga-XL. SF-AC: SF with Aho-Corasick; NE-A: named-entity accuracy; NE-WER: named-entity WER.

Model	**GigaSpeech**	**DefinedAI**		**Earnings21**	
	WER$\downarrow$	**WER**$\downarrow$	**NE-A**$\uparrow$	**WER**$\downarrow$	**NE-WER**$\downarrow$
1) Baseline	10.6	10.4	68.0	14.4	49.2
2) SF+Transf.-LM	10.6	10.2	69.3	14.9	49.9
3) SF+n-gram LM	10.6	10.2	68.2	16.8	48.4
4) SF-AC+n-gram LM	10.6	**10.0**	70.0	**12.9**	45.3
5) SF-AC+bias-list	-	10.4	**77.9**	16.7	**38.4**
6) SF-AC+4)+5)	-	**10.0**	73.3	13.1	42.3

text data from train sets. Each Transformer model is trained for 10 epochs and has around 38M params.[26]

Evaluation protocol In addition to the word error rate (WER) metric, we evaluate the accuracy and WER only on NEs: *NE-A* and *NE-WER*. NE metrics are calculated after the reference and hypothesis alignment and only strings containing NEs are taken into account. Accuracy is calculated binary: "correct" – when the NE is completely recognized correctly, "incorrect" – when at least one word-level error occurs within the NE. For evaluation on Earnings21, we use the *fstalign tool*[27] and for NEs we used only "PERSON" and "ORG" categories. To measure OOV words recognition, we choose the character error rate (CER) metric because we believe it can better reflect the model's performance when relying primarily on acoustic data. Finally, RTFX is measured on the DefinedAI test set with one RTX 3090 GPU.

3.4.5 Results & Discussion

SF-AC of n-gram LM To distinguish between out-of-domain and in-domain performance, we present results on DefinedAI and Earnings21 (Table 3.11) separately from CommonVoice (Table 3.12 and Figure 3.6). For both setups, the results of fusion n-gram LM with AC trie lead to relative WER reduction w.r.t beam search alone: 3.8% for DefinedAI, 10.4% for Earnings21, 1.5%, 1.3%, 2%, and 2.6% for EN, DE, FR, and ES from CommonVoice respectively. Moreover, for the out-of-domain sets, it improves the performance compared to the fusion with Transformer-LM: i.e., from 10.2 to 10.0 for DefinedAI and from 14.9 to 12.9 for Earnings21. In addition to improved performance, training n-gram LM is fast and easy and can be done even with a small corpus (e.g., 50h in the case of DefinedAI), which will benefit low-resource scenarios.

SF-AC of n-gram LM+keywords The biggest improvement on NEs is always achieved with

[26] Transformer-LM recipe: https://github.com/k2-fsa/icefall/tree/master/icefall/transformer_lm

[27] Provided by authors of [185] as the dataset references are in a special NLP-format: https://github.com/revdotcom/fstalign

Table 3.12: WERs of biasing techniques for Transducer models trained on 4 languages of CommonVoice. SF-AC: SF with Aho-Corasick. Our models are competitive or even outperform Whisper.

Model	**EN**	**DE**	**FR**	**ES**
Previous work				
Whisper-S (244M) [188]	14.5	13.0	22.7	10.3
Whisper-M (769M) [188]	11.2	8.5	16.0	6.9
Ours (Zipformer - 70M params)				
1) Baseline	13.5	7.7	10.0	7.8
2) SF+Transf.-LM	13.3	7.6	**9.8**	7.7
3) SF+n-gram LM	13.3	7.6	**9.8**	**7.6**
4) SF-AC+n-gram LM	13.3	7.6	**9.8**	**7.6**
5) SF-AC+bias-list	13.7	7.7	9.9	7.8
6) SF-AC+4)+5)	**13.2**	**7.4**	**9.8**	**7.6**

keyword biasing: 14.6% and 22% for DefinedAI and Earnings21 of relative improvement in accuracy w.r.t baseline (Table 3.11). Yet, the overall WER does not improve or even degrades, e.g., in the case of Earnings21. The overall WER is improved if keyword biasing is combined with n-gram LM: relative improvement w.r.t keyword biasing alone is by 3.8%, 21.6%, 3.6%, 3.9%, 1%, and 2.6% for DefinedAI, Earnings21, and CommonVoice EN, DE, FR, and ES respectively. It is also important to note that different datasets have different numbers of NEs and utterances that contain NEs (see Table 3.10). This explains the lowest impact of fusion on the WER for FR and ES, i.e., they have the least number of NEs.

Figure 3.6 illustrates the difference in the NE accuracy between different methods, where the most notable is the comparison between *keyword biasing* VS *keyword biasing+n-gram LM*. Along with the overall improvement in WER, combining n-gram LM with keyword biasing results in some degradation in NE recognition compared to keyword biasing alone. When an n-gram LM is added, the overall n-gram statistics change making it more difficult to promote specific key entities. A similar tendency is observed for the DefinedAI and Earnings21 datasets.

RTFX The RTFX results in Table 3.13 show that decoding with *keyword biasing+n-gram LM* is slightly slower compared to beam search [43] alone[28] and thus it can be used on-the-fly: RTFX of the *keyword biasing+n-gram LM* method (experiment (5)) is 6.0% lower than the baseline (1) and there is no degradation w.r.t keyword biasing (4). Although the WER performance of n-gram LMs SF "with" VS "without" AC-algorithm is the same on all the test sets (compare the 2nd and the 3th experiment rows in Table 3.11 and 3.12), decoding with SF-AC ((3) in Table 3.13) is 31% faster than with SF without AC (2): 77.8 RTFX against 111.1.

[28] See code in k2/Icefall in: https://github.com/k2-fsa/icefall/blob/master/egs/librispeech/ASR/pruned_transducer_stateless2/beam_search.py.

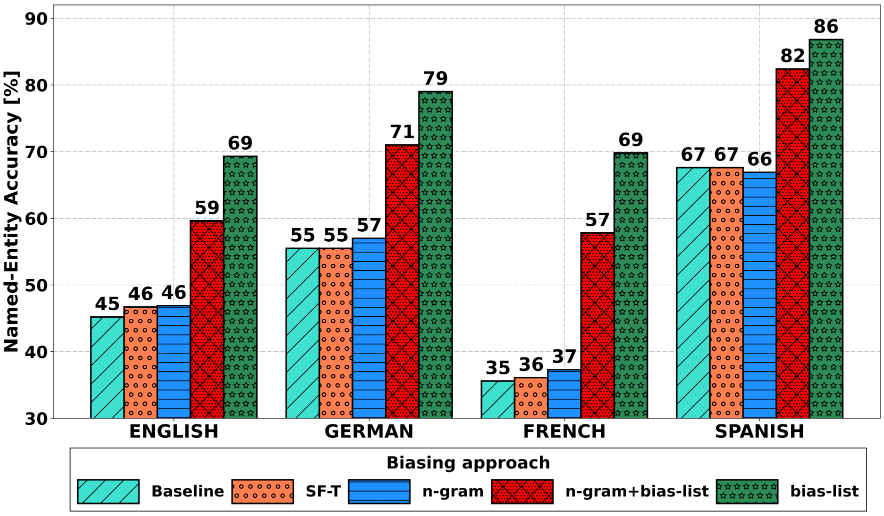


Figure 3.6: NE accuracy for different approaches on 4 languages of CommonVoice. Including a list improves NE accuracy while marginally decreasing performance when adding n-gram LM.

Table 3.13: Ablation of decoding speed (RTFX; higher, better) and character error rate (CER) on OOV words with SF. SF-AC: SF with Aho-Corasick. Note that SF-AC+n-gram LM+bias-list improves CER with a negligible decrease in RTFX.

Model	**Defined-AI**	
	oov-CER↓	**RTFX↑**
1) Baseline	35.0	120.7
2) SF+n-gram-LM	36.2	77.8
3) SF-AC+n-gram LM	34.2	111.1
4) SF-AC+bias-list	32.9	113.5
5) SF-AC+3)+4)	**32.6**	117.3

OOV words The word level statistics from n-gram LM do not lead to any degradation of recognition of OOVs on the subword level (Table 3.13). The n-gram LM fusion with the AC algorithm improves by 2.3% over the baseline and 5.5% over the n-gram LM fusion without AC. This can be explained by the ability of AC to find partial matches. The best OOV performance with *keyword biasing* and *keyword biasing+n-gram LM* methods is because the DefinedAI bias list includes some OOV words.

Conclusion We demonstrate the benefits of integrating n-gram LMs with a Transformer-Transducer model during decoding with SF and an Aho-Corasick-based trie. The n-gram LM weights are loaded into the trie and during decoding the Aho-Corasick string matching algorithm is used. This leads to significantly faster decoding than SF without AC, on-par WERs overall, and no loss in decoding time (RTFX) with regard to shallow fusion alone.

3.5 Using Contextual Knowledge at ASR Training Time

In this final section, we propose a two-step approach to add contextual knowledge during semi-supervised training to reduce WERs on the parts of the utterance that contains the callsign, a named entity. Initially, we represent in a WFST the contextual knowledge (i.e., air-surveillance data) of an ATCo-pilot communication. Then, during Semi-Supervised Training (SST) the contextual knowledge is added by second-pass decoding (i.e., lattice re-scoring). Results show that 'unseen domains' are further aided by contextual SST when compared to standalone SST. For this task, we introduce the Callsign Word Error Rate (CA-WER) as an evaluation metric, which only assesses ASR performance of the spoken callsign in an utterance.

Publication Note

The material presented in this section is adapted from the following publication:

- J. Zuluaga-Gomez, I. Nigmatulina, A. Prasad, P. Motlicek, K. Veselý, M. Kocour, and I. Szöke, "Contextual Semi-Supervised Learning: An Approach to Leverage Air-Surveillance and Untranscribed ATC Data in ASR Systems," in *Proc. Interspeech*, 2021, pp. 3296–3300

Supplementary materials related to this section:

- ATCO2 project website: https://www.atco2.org/

Major contributions Problem definition and experimental design and setup. Data preparation. Trained the hybrid-based ASR systems for the experiments. Lead the work, including the paper write up.

3.5.1 Introduction

In this section, we introduce the usage of contextual knowledge at training time of Hybrid-based ASR systems. This section differs from Section 3.3 and Section 3.4, as both focus only on boosting information at decoding time. Thus, the main contribution of this section is how we can leverage large amounts of untranscribed data with contextual knowledge to improve the gains w.r.t semi-supervised learning alone.

Current commercial ASR systems are trained on thousands of annotated audio-text pairs, whereas in the ATC domain not even a considerable fraction of that amount is available for supervised training. Recent research on ASR in ATC has concluded that the lack of annotated speech data and its high production cost are current issues holding the development of fully autonomous ASR systems [112]. Some previous research addressed the lack of transcribed ATC speech data using SST (e.g., ASR tasks applied to under-resourced languages [189, 138, 190]) to decrease WERs [114, 115]. Here, we investigate the effect of integrating contextual knowledge from air-surveillance data into the SST pipeline to further boost the performance w.r.t SST alone. Similar research adding contextual knowledge into the decoding graph (HCLG.fst) or by re-scoring lattices after the decoding step were described in [191, 169, 168, 192]. Modifying the Language Model (LM) with prior knowledge is reviewed in [193, 194]. Contextual biasing for hybrid-based

ASR is covered in the thesis in Section 3.3 and Section 3.4 for E2E models.

3.5.2 Contextual Semi-Supervised ASR Training

An ATCo-pilot communication heavily relies on the very particular context they are in. Characteristics such as airplane location, altitude, departure or arrival, and air-space status define the information that could be uttered by the speakers (small deviations are allowed in specific scenarios). For instance, an ASR system can leverage this particular contextual information (mentioned above) as prior knowledge to increase its performance. However, aspects such as speaker's characteristics, location and context, low SNR levels, and air-space status increase the challenge of ASR for the ATC task.

Contextual adaptation in ASR Our work relies mostly on adding air-surveillance data as contextual knowledge in the ASR system, also known as 'contextual ASR'. Contextual ASR has been an active topic of research in the last decade, where companies such as Google and Microsoft have leveraged contextual data (e.g., user location and contact list) for boosting mobile devices' ASR performance. One of the straightforward ways of adding context into the system is by biasing the LM. See Section 3.3 for further information about the lattice re-scoring and G-boosting (or G-extension). See Section 3.3 for further information.

Contextual ASR in air-traffic control communications The ICAO is the entity that regulates the phraseology and grammar used in ATCo-pilot voice communications. A standard communication starts with a callsign, followed by a command, and a value. One of the main challenges in ATC (thus in ASR) is to correctly identify the sequence of words in the utterance that denotes the callsign, which specifically addresses an individual aircraft. This research focuses on using a list of callsigns as prior knowledge in the ASR system to reduce the search space, thus increasing overall recognition performance. Previous work has attempted to incorporate contextual knowledge in the recognition process [165, 166, 164, 167]. We redirect the reader to a general review about spoken instruction understanding in the ATC domain to [111]. Nevertheless, most of the previously cited works in ASR for ATC employ only data from few airports assuming high-quality speech, i.e., high SNR $\sim$20dB. Despite this, it is hard to determine the quality of ATC speech in advance due to external elements, e.g., weather, cockpit or environmental noise.

Semi-supervised training in ASR SST has been proven to be an important asset for ASR in many tasks. The goal of SST is to leverage large amounts of non-annotated (i.e., data augmented with automatically generated transcripts) data to boost the performance of the ASR trained in a supervised manner. There have been many recent studies leveraging untranscribed data during ASR training; for example, pre-training and self-training methods in end-to-end ASR systems [195]. Other research has leveraged non-annotated data for ASR in low-resource languages [139]. Regarding ATC voice communications, previous researchers have explored different techniques

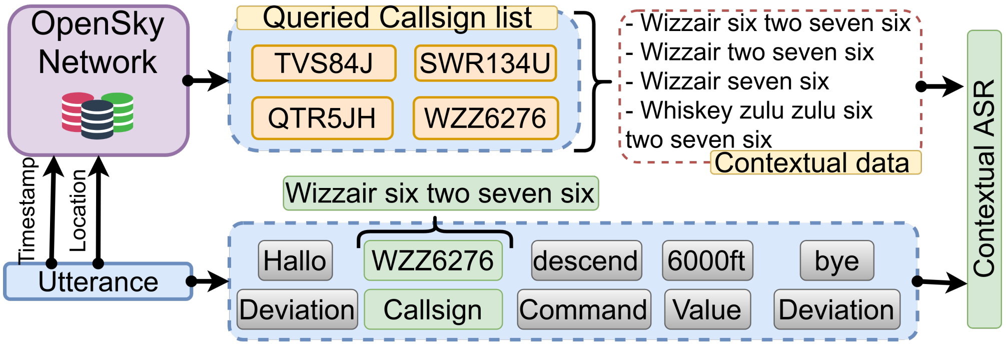


Figure 3.7: Process of retrieving a list of callsigns (contextual data) from OpenSky Network. This is the list of all possible verbalization of each callsign.

for leveraging untranscribed ATC data with SST [114, 115].

3.5.3 Databases and Experimental Setup

Database We use a mix between supervised and untranscribed data. The set of supervised ATC databases contains a mix of private and public corpora, as referenced in Table 2.2. [29] On the other hand, there are several ways to obtain untranscribed ATC speech data. For this study we gathered data from two sources that rely on VHF receivers: i) open-source channels such as LiveATC[30], and ii) recordings from low-to-mid-quality VHF receivers offered by the contributors from the ATCO2 project. The recording quality is proportional to the placement of the VHF receiver (close or far from an airport, surrounding environment or altitude of the plane) and the quality of the hardware itself. First, we manually transcribed 1.9h of recordings (mostly noisy speech) from LiveATC to assemble a challenging test set. We tag it as '*liveatc_mix*' including recordings from EIDW, LSZH, KATL, EHAM, ESGG, and ESOW airports. The SNR levels for *liveatc_mix* test set ranges from 5-15 dB. Secondly, we gathered 67h (49 thousand segments) of ATCo-pilot speech with high-quality setups of VHF receivers in Prague (LKPR) and Brno (LKTB) airports from August 2020 until January 2021. We tag it as '*unsup_vhf_67h*' untranscribed train set. We annotated 5 minutes (without silences) of speech collected with VHF receivers from Brno airport (not present in the supervised data), i.e., '*aiport_lktb_vhf*' test set. Additionally, we automatically extract timestamp and location information for each utterance in *unsup_vhf_67h* to extract callsigns list, as listed in Figure 3.7.

Integrating contextual knowledge in semi-supervised training Currently, all the airplanes circulating in Europe must be equipped with Automatic Dependent Surveillance–Broadcast (ADS-B) and Mode S modules which transmit almost in real-time their information as meta-data

[29] (We use, ATCOSIM, UWB-ATCC, LDC-ATCC, MALORCA, and AIRBUS.

[30] LiveATC.net is a streaming audio network consisting of local receivers tuned to aircraft communications: https://www.liveatc.net/

such as altitude, velocity, callsign, and direction. OpenSky Network[31] stores ATC information through feeding ADS-B data from network of feeders placed all around the world, and collecting ADS-B data through receivers (similar to VHF voice communication). This data can be retrieved by defining a query. We define a query based on the utterances' timestamp and scanned area (*unsup_vhf_67h* untranscribed set). OSN retrieves a list of callsigns in ICAO format for each utterance that match the query criteria (potentially one callsign from this list is present in the given utterance). However, our ASR system is trained with transcripts that have the verbalized version of the callsigns instead of ICAO format. We developed an algorithm that verbalizes the ICAO callsigns into different versions. The process is then repeated for each callsign from the callsign list. Figure 3.7 shows the pipeline to assemble the contextual data from the verbalized callsign list for one utterance. Finally, we repeat this pipeline for each utterance of the unsupervised train set, *unsup_vhf_67h*.

Verbalizing a call-sign Our previous work can give a more in-depth idea on how the list of callsigns are retrieved and verbalized [18, 22]. An example of this process for the call-sign ICAO code: `TVS123AB` is:

```
skytravel one two three alfa bravo
skytravel three alfa bravo
skytravel alfa bravo
skytravel one alfa bravo
skytravel one two bravo
tango victor sierra one two three alfa bravo
one two three alfa bravo
three alfa bravo
alfa bravo
```

Baseline ASR system The lexicon is composed of a word-list assembled from the transcripts of all available annotated train databases and from additional public resources (e.g., airlines names, airports, countries, ICAO alphabet, way-points, etc.). The pronunciation of new words is obtained with Phonetisaurus G2P [143]. The language model is a tri-gram LM created by interpolating several LMs. An additional LM (only used during the interpolation, to further tune the final LM) is built from external data such as expanded callsigns from 2019,[32] expanded runaways (all combinations) and European way-points. All experiments are conducted with Kaldi speech recognition toolkit [100]. We report results with hybrid-based ASR systems. The models are composed of six convolution layers and 15 factorized time-delay neural network, i.e., CNN-TDNNF. We use the standard chain Lattice-free MMI (LF-MMI) based Kaldi's recipe [153]

[31] OpenSky Network: provides open access of real-world air traffic control data to the public.

[32] Website URL - Crowdsourced air traffic data from The OpenSky Network 2020: https://zenodo.org/record/3901482

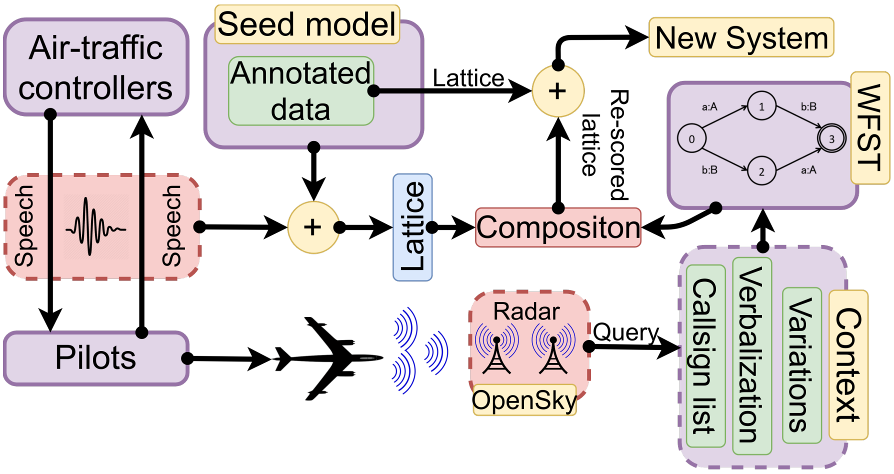


Figure 3.8: Contextual semi-supervised training pipeline.

for training the seed and SST-based models. It also requires 100-dimensional i-vector features and 40-dimension MFCC features. We triple the training data by adding noises between 5-10dB SNR and then between 10-20dB SNR. The baseline ASR system is trained on all available data for 5 epochs, denoted as 'seed system'.

Contextual semi-supervised training We follow the standard recipe for SST [139], where a seed system produces word recognition lattices of the untranscribed data set (e.g., *unsup_vhf_67h*), which are then mixed with the lattices generated on manually transcribed data to train a new acoustic model. In hybrid ASR, lattices are representations of search results that act as 'intermediate format' that contain timing information with more details than plain 1-best string or n-best lists. Lattices generated on manually transcribed and untranscribed data are mixed and a new model is trained with this merged data. There are several ways to add contextual knowledge in the ASR system, e.g., tuning LM towards a defined sequence of n-grams, modifying *G.fst* when making HCLG graph, or simply re-scoring lattices. This research only explores lattice re-scoring during SST. Initially, we create a Weighted Finite-State Transducer (WFST) graph for each utterance in the untranscribed dataset (i.e., *unsup_vhf_67h*). The WFST is constructed from n-grams of the verbalized callsign list (air-surveillance data retrieved from OSN). Afterward, the baseline lattices of *unsup_vhf_67h* (generated during the first pass decoding) are composed with its particular callsign WFST in a second pass decoding (see Section 3.3). The re-scored lattices are then used to retrain the acoustic model again, as presented in Figure 3.8. In the lattice re-scoring approach, lattices' weights are re-scored to increase the probability of given callsign sequences. The expanded callsigns (represented in WFST) get boosted during the re-scoring process, thus they become more probable to appear in the hypothesized transcripts.

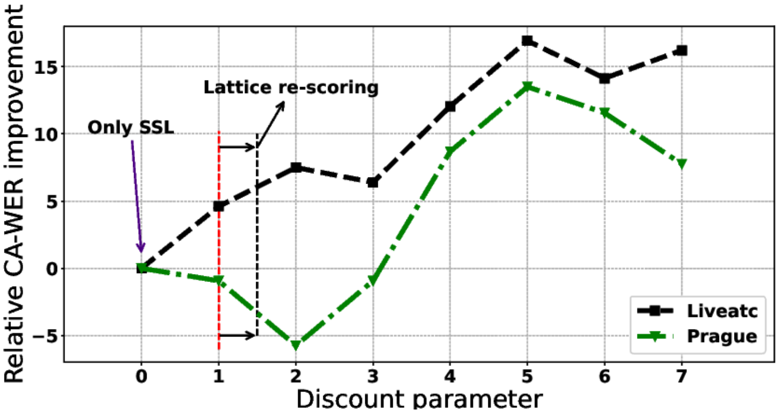


Figure 3.9: CA-WER performance on liveatc_mix (noisy) and Prague (clean) test sets for different discount parameters used at the moment of creating the biasing WFST.

3.5.4 Results & Discussion

We perform four different experiments to test the inclusion of contextual knowledge in SST. First, we train a baseline AM (i.e., seed model) without SST (first row of Table 3.14). Then, we train a new acoustic model from scratch with SST, the seed model is used to generate the lattices of the untranscribed data set (*unsup_vhf_67h*). Next, we re-scored the untranscribed data lattices by composing them with the WFSTs (one for each utterance) previously created. The lattice re-scoring approach relies on a 'discount' hyperparameter, which tells how much weight is given to the 'contextual knowledge' encoded at the moment the WFST is created. We report the last result on using a discount parameter of 6.0 instead of 2.0. SST gave much larger improvement for test sets that matched the data used in semi-unsupervised learning (i.e., similar SNR and airport location). For example, we obtained around ∼20% relative WER improvements in *liveatc_mix* and *aiport_lktb_vhf* test sets, and 13.6% relative WER improvement in Prague test set by doing standalone SST. Nevertheless, Airbus and Vienna test sets show a WER degradation. We attribute this to data-quality mismatch (i.e., the untranscribed VHF data is noisier than the data with manual transcripts), but also the Airbus and Vienna test sets are from airports not present in the untranscribed set. It is important to mention that WER improvements in challenging test sets such as *liveatc_mix* and *aiport_lktb_vhf* are more significant because the data is nosier and some airports are not present in the annotated train set; which is closer to a real-life scenario. An extra ∼5% relative WER improvement is achieved on *liveatc_mix* and Prague test sets when adding contextual knowledge into the SST pipeline. The Prague test set yielded improvements in WER in all four proposed ASR systems. We believe this is because data was present in both, the transcribed and untranscribed training sets.

The WER metric measures the ASR performance in the whole utterance, however, our contextual SST approach only 'boosts' the words that belong to a callsign in the hypothesis. For instance, this process increases the probability of recognizing the correct callsign in the ATCo-pilot

Table 3.14: WERs of multiple ASR systems for different test sets. The default discount parameter (DP) in ASR systems with lattice (lat.) re-scoring is 2.0.

System	**liveatc_mix**	**aiport_lktb_vhf**	**Airbus**	**Prague**	**Vienna**
seed model	49.7	26.6	**11.0**	4.4	**6.8**
+SST	38.3	21.3	12.1	3.8	8.2
+lat. re-scoring	37.3	21.4	12.2	3.8	8.4
SST+lat. re-scoring (DP: 6.0)	**36.4**	**21.3**	11.8	**3.6**	8.4

communication.[33] We thus propose a new metric: Callsign WER '*CA-WER*' which is more aligned to measure the ASR system performance only on callsigns, between the reference and hypothesized text. We use *texterros*[34] library to evaluate CA-WER, which needs the verbalized ground truth callsign per utterance. We evaluated CA-WER for *liveatc_mix*, Prague, and Vienna test sets; 610, 875, and 915 utterances have a callsign, respectively. The CA-WER is evaluated for different discount parameters (hyperparameter in the WFSTs). Figure 3.9 shows that lattice re-scoring helps in all cases for *liveatc_mix*, and it helps Prague test set after a discount value of 4.0. Vienna test set is skipped from Figure 3.9, because there were no significant variations across different discount parameters. Even though there is a degradation in WER for Vienna test set when adding contextual knowledge, we obtained 7.5% relative CA-WER improvement when comparing it with the '*+SST*' model (thus showing the robustness of the proposed approach). Discount parameter of 5.0 is best, i.e., 17.5% and 14% CA-WER relative improvement on *liveatc_mix* (CA-WER: 39.88% → 32.9%) and Prague (CA-WER: 3.48% → 2.99%) test sets, respectively.

Conclusions In this section (similar to Section 3.3), we introduce an SST-based approach that leverages contextual knowledge. It relies on ATC speech and air-surveillance data as input modalities. First, we create a biasing WFST for each utterance, that encodes n-grams sequences of verbalized callsigns retrieved from OpenSky Network. This prior knowledge in the format of WFST is then added into the SST recipe to further improve the acoustic models.

[33] The callsign is composed of five to seven words, i.e., 25% of the transcript.

[34] https://github.com/RuABraun/texterrors

3.6 Building ASR Systems for ATC

This section gather the main solutions covered in Chapter 3 and propose some directions to build reliable ASR systems for ATC communications.

- **Step 1** Supervised data is imperative for training custom ASR systems for ATC applications. There are multiple open source ATC databases (see Table 2.2), albeit of small size w.r.t databases sizes for current E2E models. The first step is that we propose to acquire the ATCO2 corpora which contains more than 5000h of pseudo-labeled ATC audio with a robust in-domain hybrid-based ASR system developed by ATCO2. This data has proven to be reliable to train hybrid-based and E2E systems from scratch and without other source of supervised data, see [3]. Moreover, the lessons learned from ATCO2 data annotation pipeline [5] can be followed if more pseudo-labeled data is required for a specific use case;
- **Step 2** It is important to have good transcriptions practices for unifying multiple open-source ATC databases. In [196], authors have developed custom ontologies and practices for ATC annotation. During the ATCO2 project [5], we released a cheat-sheet to provide guidance on how to annotate ATC speech [4]. We recommend following these practices in order to minimize confusion due to misspelling of certain specific words, e.g., callsigns;
- **Step 3** Developing robust ASR systems for ATC communications is imperative. We propose multiple solutions on how to develop ASR systems for both, hybrid-based and E2E ASR. Note that depending on the amount of supervised data available at the training stage, the latency, and the performance requirements, one could choose one of these architectures. However, we propose the usage of pre-trained E2E systems, as they have shown lower WERs, even on low-resource settings;
- **Step 4** ATC surveillance data is an appropriate source of real-time contextual information that can be used to improve ASR outputs. Its integration can lead to substantial benefits in terms of WERs and improved callsign recognition rates. In this chapter, we cover how to integrate contextual information for both, hybrid-based and E2E models;
- **Step 5** Multiple downstream task are required in real-life ATC applications. We propose a variety of solutions for NLU, speaker role detection and text-based speaker diarization. These are cover in Section 5.1 and Section 5.2. Furthermore, NLU tasks are of special interest in the ATC community because the high-level information can be used to assist ATCos in their daily tasks, thus, reducing their overall workload.

4 Speed and Compute Bounded Low-Resource Speech Recognition

Introduction

In this chapter, we address a variety of challenges when developing ASR for real-life use cases. We propose (1) a method for ASR prototyping with pseudo-labeled data from foundational speech models (§ 4.1); (2) a self-supervised-based encoder ported to the transducer architecture (§ 4.2) and (3) how to improve its WERs on low-latency settings § 4.3; and finally (4) HyperConformer, a new architecture that achieves comparable or higher recognition performance w.r.t Conformer while being more efficient than Conformer in terms of inference speed, memory, parameter count, and available training data (§ 4.4). An overall overview of this chapter is in Figure 4.1.

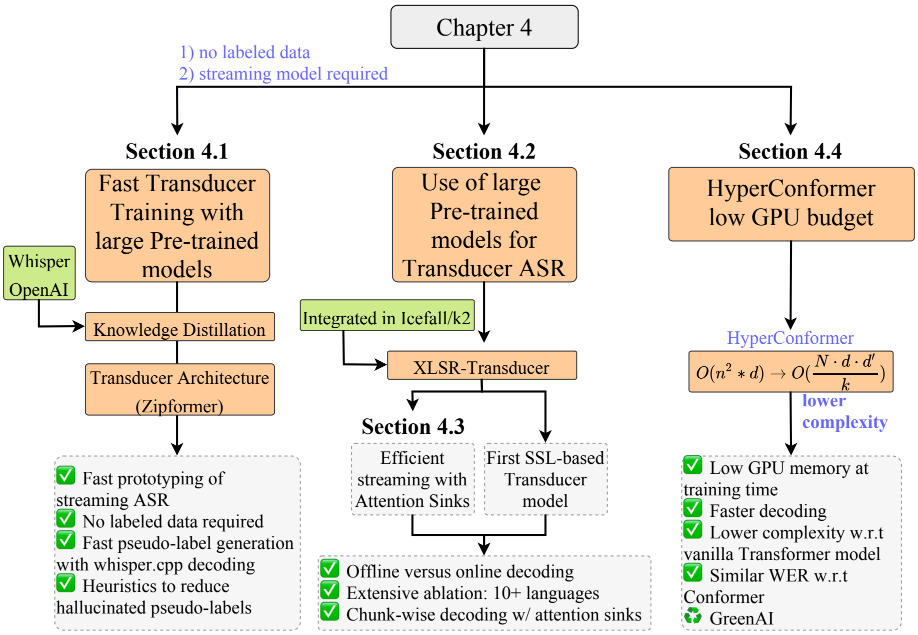

Chapter 4
1) no labeled data
2) streaming model required
Section 4.1
Fast Transducer Training with large Pre-trained models
Whisper OpenAI
Knowledge Distillation
Transducer Architecture (Zipformer)
✓ Fast prototyping of streaming ASR
✓ No labeled data required
✓ Fast pseudo-label generation with whisper.cpp decoding
✓ Heuristics to reduce hallucinated pseudo-labels
Section 4.2
Use of large Pre-trained models for Transducer ASR
Integrated in Icefall/k2
XLSR-Transducer
Section 4.3
Efficient streaming with Attention Sinks
First SSL-based Transducer model
✓ Offline versus online decoding
✓ Extensive ablation: 10+ languages
✓ Chunk-wise decoding w/ attention sinks
Section 4.4
HyperConformer low GPU budget
HyperConformer
$O(n^2 * d) \rightarrow O(\frac{N \cdot d \cdot d'}{k})$
lower complexity
✓ Low GPU memory at training time
✓ Faster decoding
✓ Lower complexity w.r.t vanilla Transformer model
✓ Similar WER w.r.t Conformer
♻ GreenAI

Figure 4.1: Overview of Chapter 4.

4.1 Fast Transducer ASR Prototyping with Pretrained Models

The training of automatic speech recognition (ASR) with little to none supervised data stills remains an open question. In some cases, this involves a pre-train then fine-tune stage that requires large data and computational budget. In this work, we demonstrate that Transformer transducer (TT) models can be trained from scratch in consumer and accessible GPUs in its entirety with pseudo-labeled (PL) speech from foundational speech models (FSM). We perform a comprehensive ablation on different aspects of PL-based TT models such as (1) offline decoding, (2) chunk-wise decoding for low-latency streaming applications, and (3) TT final WER as the function of the FSM size. Our results, demonstrate that TT can be trained from scratch without supervised data, even with very noisy PLs. We validate the proposed methodology on more than six languages from CommonVoice and propose multiple heuristics to filter out hallucinated PLs.

Our contributions are covered below:

- Comprehensive study of pseudo-labels quality on downstream TT ASR models;
- impact of PL quality on offline versus online settings;
- robust heuristics to filter out noisy and hallucinated PLs from Whisper;
- study of the impact of FSM model size on the final WERs of the TT models.
- multiple ablations, including mix in of supervised data as regularization during training;
- validation of the proposed approach on six languages from CommonVoice.

Publication Note

The material presented in this section is adapted from the following publication:

- I. Nigmatulina, J. Zuluaga-Gomez, *et al.*, "Fast Streaming Transducer ASR Prototyping via Knowledge Distillation with Whisper," in *Submitted to EMNLP 2024 (long paper).* ***[Equal contribution]***, 2024

Major contributions Problem definition and experimental design and setup. Data preparation for CommonVoice. Trained the Transducer ASR systems for the experiments. Lead the work, including the paper write up.

4.1.1 Introduction

There are many challenges when developing ASR for industrial applications, including (1) required large scale databases that generalize across multiple domains; (2) inference under challenging low-latency settings; and (3) lightweight and reduced number of parameters to minimize deployment costs. While the first has been solved by training large acoustic foundational speech models (FSM) with massive databases [51, 80], the latter two strongly relate to architectural choices, e.g., using Connectionist Temporal Classification (CTC) [45] or transducer-based [53] modeling.

In industrial applications, large-scale supervised databases in the target domains are not always

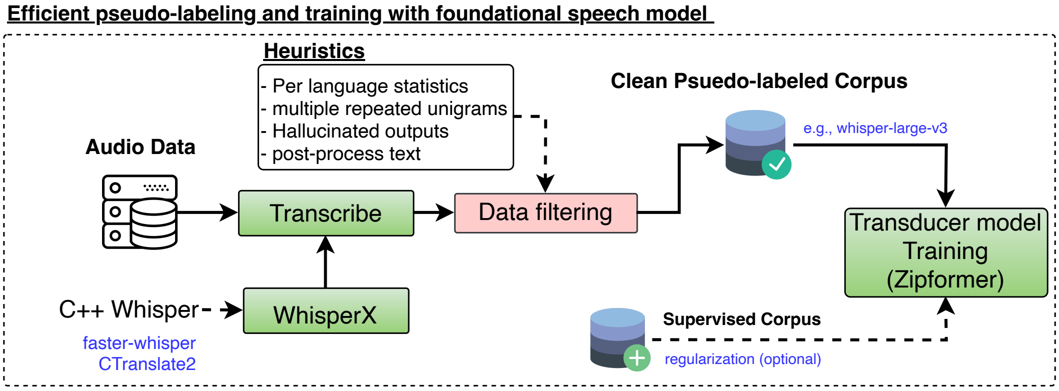

Efficient pseudo-labeling and training with foundational speech model
Heuristics
- Per language statistics
- multiple repeated unigrams
- Hallucinated outputs
- post-process text
Clean Psuedo-labeled Corpus
Audio Data
e.g., whisper-large-v3
Transcribe
Data filtering
Transducer model Training (Zipformer)
C++ Whisper
WhisperX
faster-whisper
CTranslate2
Supervised Corpus
regularization (optional)

Figure 4.2: Proposed approach for efficient psuedo-labeling with Foundational Speech Models.

available, thus several techniques have been proposed to develop robust ASR models with small corpora, e.g., (1) data augmentation [160, 197]; (2) only-audio self-supervised pre-training with large databases and then fine-tuning with small corpora [13, 49, 51] (3) pseudo-label then fine-tune, e.g., semi-supervised learning [17, 198, 199]. Most of these approaches target the attention-based encoder-decoder (AED) [70] or CTC-based models. Even though these two architectures have shown strong results on multiple benchmarks (e.g., Whisper [188]), they still lag behind in the streaming setting [200].

For industrial use cases, which requires low-latency streaming decoding, the Transducer [53] have been explored as it naturally supports this configuration [64, 66]. However, TT models are not as popular as AED or CTC because they were harder to train, until it was shown that they can attain WERs akin to AED models [54]. The transducer models consist of an encoder, predictor and joint networks. Using Transformer [61] encoder, lead to Transformer Transducer (TT) [62, 63]. They are trained from scratch, thus requiring large-scale supervised datasets [64, 65] in the target language and domain.

In this work, we focus on two questions partly unanswered by the research community: (1) Could we prototype a streaming Transformer-Transducer ASR model of sufficient quality on the target domain with consumer-level GPUs? (2) Can we train TT models with pure pseudo-labeled (PL) data? We target the streaming scenario, which is by nature more challenging than standard offline (full attention) decoding [54]. Despite the robustness of AED models on the offline scenario, they still require large amount of supervised data. Still, there are works that aim to bring them to streaming setups [11]. Here, we use TT models [62], where the challenge arises on the fact that these do not include a self-supervised stage,[1] i.e., needing always audio text pairs. We show that TT models can be trained from scratch in its entirety with PLs from Whisper [188] while attaining competitive WERs on the streaming scenarios. The overall proposed approach is in Figure 4.2.

[1] [201] explore to warm start the encoder with a pretrained SSL-based model, albeit closed source model.

4.1.2 From Encoder-Decoder to Transducer ASR

The advantage of transducer models over encoder-decoder relies on the fact that it supports streaming decoding. Only recently, it was demonstrated that these models can surpass standard AED ASR models [54]. There have been multiple breakthroughs that have made possible the training of transducer easier, such as (1) pruned transducer loss [60], (2) better architectures, e.g., FastConformer [59, 65] and HyperConformer [12]; and (3) from the modeling side, e.g., model pruning and sparsifiation [202] and quantization [54]. However, little to none work has been done on fast TT prototyping ($\sim$1 GPU-day fixed compute) with pure pseudo-labeled data.

4.1.3 ASR Pseudo Labeling

Semi-supervised learning [203], pseudo labeling [204], and weakly supervised learning [188] are a family of methods aiming to partly alleviate the burden of lack of labeled data for supervised ASR training. These approaches have shown successful WERs on multiple settings and languages. In practice, a teacher model g is trained on an audio-text paired corpus $D_l = \{X_i, Y_i\}$, then it is used to pseudo label a much larger unlabeled only audio corpus, $D_{pl} = \{X_i, Y_i^*\}$. Then a smaller model [79] can use D_l and D_{pl} for supervised training or fine-tuning [205]. However, PL are often noisy and bounded by g model quality, whereas its use might result in suboptimal final WERs in the models. This can be solved by either filtering out the most noisy samples or increasing g model size to increase their quality.[2] Several approaches to improve the PL quality, includes improving the loss functions [198, 206], pairing online and offline models at training time [203], and by continuous single-language [207, 208] and multilingual pseudo-labeling setting [199].

4.1.4 Knowledge Distillation with Large Models

Knowledge distillation (KD) is a very well-known technique to distill the knowledge from a large model into a small model [209]. The former is regarded as the *Teacher* while the latter as the *Student*. In this framework, we first train the teacher model with the correct label (for supervised training) [210] or in a self-supervised manner. Then, the student model is trained with the posterior distributions of the pretrained teacher model [211]. The KD setting is also known as teacher-student training [212]. There has been prior work on KD for CTC [210] and AED models with Whisper [188] in [213, 214] and Transducer models [215]. Another line of work has aimed at KD from offline to online transducers into [216] or by using self-supervised models as teachers [217].

In this work, we focus on sequence-level KD, which means that we use the 1-best hypothesis obtained from the teacher model instead of using the posterior distribution. This has some benefits, as the expenses of trading off flexibility, e.g., (1) no need to cache the teacher model

[2] Here we assume that a larger model attains higher quality, i.e., lower WERs.

or its outputs into memory; (2) no need to modify the current ASR training pipelines; (3) faster ASR training w.r.t teacher-student based KD; and (4) we can use highly optimized only inference pipelines–that support model quantization–for PL generation, e.g., WhisperX [218], which can lead to faster development.

4.1.5 Databases and Experimental Setup

Our core contribution is the fast prototyping of TT ASR trained with pure pseudo-labeled data that can work on streaming low-latency settings. We select the Whisper model as our teacher model [188] due to its strong performance across benchmarks in multiple languages. In addition, it provides models at different parameter scales, opening the door to studying the effect of how PLs from different qualities impact downstream TTs models.

Pseudo labeling with WhisperX We use the WhisperX pipeline [218] across all the experiments to generate PLs. It is composed of a (1) voice activity detection step to segment long-form audio; (2) batching multiple segments for efficient inference; (3) model quantization of Whisper and C++ implementation on FasterWhisper[3] which uses CTranslate2 for fast decoding;[4] (4) model inference and word level alignment. Note that we pseudo-label each training corpus with 5 Whisper model sizes, i.e., tiny, base, small, medium, and large-v3.

Table 4.1: Maximum number of characters allowed in each pseudo-labeled word with Whisper.

Language	**CA**	**EN**	**DE**	**FR**	**ES**	**IT**
Max. characters per word	16	16	30	20	25	22

Data filtering heuristics We developed multiple data selection heuristics (H) to filter-out noisy and hallucinated PLs:

- $H1$**:** remove PL if composed of the same unigram three or more times.
- $H2$**:** compute maximum word length from supervised training corpus and removed utterances with one or more PLs larger than the max threshold. See Table 4.1 for the exact statistics per language. Note that languages that join words, such as German (DE) has a substantially larger threshold.
- $H3$**:** compute $word_{ratio}$[5] and filter out samples with $word_{ratio}$ less than 1 or more than 4.[6]
- $H4$**:** verbalize all the numbers from the pseudo-labels, remove punctuation and normalize following the CommonVoice recipe in LHotse [219].

[3] https://github.com/SYSTRAN/faster-whisper

[4] https://github.com/OpenNMT/CTranslate2/

[5] Number of words divided by utterance duration [seconds].

[6] The WhisperX pipeline uses a VAD system to remove silences in the audio. In our case, CommonVoice utterances are already pre-segmented, thus we omit this step.

These heuristics are applied for every training corpora. Similar heuristics are proposed in [79].

Transformer transducer training We train Transformer-Transducer models from scratch for each language from Common. We use Zipformer stateless [55] TT model [58] with the latest Icefall Transducer recipe and its default training hyper-parameters.[7] This includes *ScaledAdam* optimizer [186], learning rate scheduler with a 500-step warmup phase [61] followed by a decay phase (each 7.5k steps and 3.5 epochs), as in [58]. The neural TT model is jointly optimized with an interpolation of simple and pruned RNN-T loss [60, 53] and CTC loss [45] ($\lambda = 0.1$), according to:

$$\mathcal{L} = (1 - \lambda) \cdot \mathcal{L}_{RNNT} + \lambda \cdot \mathcal{L}_{CTC}. \tag{4.1}$$

The peak learning rate is $lr = 5.0e^{-2}$ and we train each TT for 30 epochs on a single RTX 3090 GPU with only PLs.

Regularization with supervised data We perform experiments where along with PLs we mix in 100h of randomly selected supervised data from the train set D_l during training. We compute mixing weights between D_l and D_{pl} so each training batch contains at least one sample from D_l. This is achieved with *CutSet.Mux* function from Lhotse [219].[8] All the experiments that uses PL and supervised data are denoted with ***+sup. [100h]***, otherwise, the model is trained with PL only. As an ablation experiment, we also test the performance by scaling up supervised data to 200h and 400h when using the weakest FSM, i.e., whisper-tiny. This experiment aims to (1) compensate for very low-quality PLs, and (2) demonstrate that Whisper PLs (from the largest models) are of sufficient quality for transducer training without any supervised data.

Enabling streaming decoding with multi-chunk training All the models proposed in this work can perform streaming decoding. This is achieved by performing chunk-wise multi-chunk training. During training, we use causal masking of different sizes to enable streaming decoding under different low-latency configurations [220, 11]. Specifically, we rely on two lists: chunk-size={640ms,1280ms,2560ms,full} and left-context-frames={64,128,256,full}.[9] At training time, we randomly select the chunk size and the left context chunks for each batch. This enables the final model to work on a wide variety of streaming settings. At test time, we select 13 different decoding configurations ranging from 320 ms[10] to 2560 ms chunks.

CommonVoice database We use the CommonVoice dataset, as described in Table 2.1 in Section 2.3.1. We use the following languages: Belarusian (BE), Catalan (CA), German (DE), Spanish (ES), French (FR), and Italian (IT). We use the official train sets and report WERs on the official test sets. See Table 2.1 for further statistics.

[7] https://github.com/k2-fsa/icefall/tree/master/egs/librispeech/ASR/zipformer.

[8] It lazily loads two or more datasets and mixes them on the fly according to pre-defined mixing weights.

[9] The effective number of left context chunks is computed as $left\_context\_frames//chunk\_size$.

[10] Decode chunk size of 320ms is more challenging as it has not been used during training.

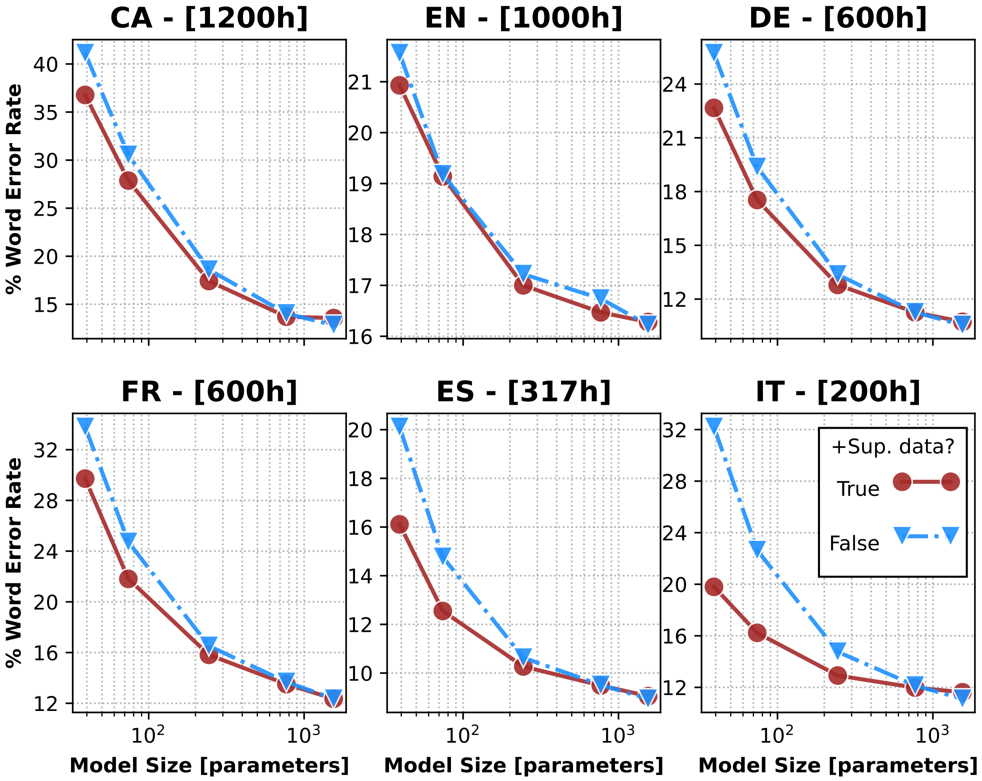


Figure 4.3: WERs for offline Zipformer models on six languages of CommonVoice. Models are trained with pseudo-labels from different Whisper model sizes (blue graphs). Adding 100h of supervised data during training (red graph) regularizes the training up to models with 700M params, especially for languages with less data.

4.1.6 Results & Discussion

We run experiments for each CommonVoice language by training streaming models in a multi-chunk fashion, i.e., we use masking during training with different configurations which allow our models to work under different configurations, as in [11, 220]. Specifically, we randomly select the chunk size from a predefined list for each batch during training.

Baseline offline models In Figure 4.3, we present the offline results for TT models trained from scratch on PL data only in six languages (depicted by blue graphs). These models are evaluated only in a non-streaming context to determine the upper bound WERs achievable by training with PLs of varying qualities. As the size of the Whisper Model increases (shown on a log-scaled x-axis), there is a corresponding improvement in WERs, also on a log-scale. The best performance is observed for ES, with the least favourable results for EN. These results show that our approach adapts across a spectrum of PL data quantities and qualities, ranging from 200h for IT to over 1000h for CA and EN. We additionally analyzed the performance of models trained on PLs depending on how well each language is represented in the data used for training Whisper

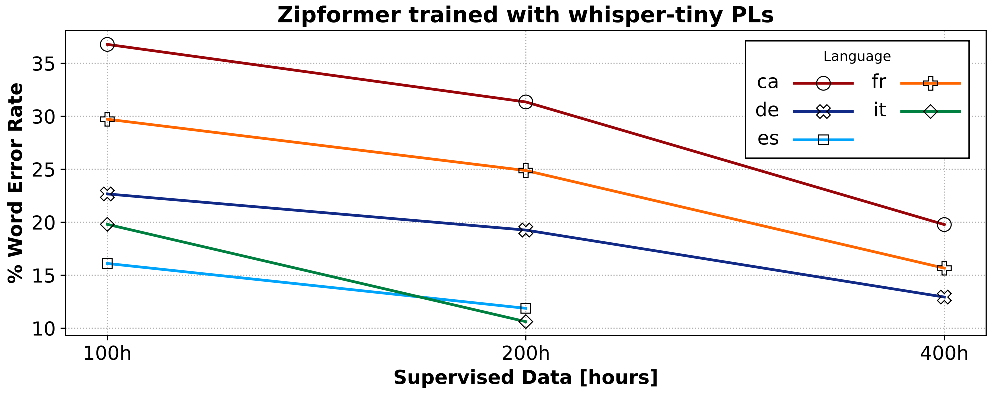


Figure 4.4: Ablation of impact of mixing in supervised data during training with very weak pseudo-labels. WERs for multiple Zipformer models trained from scratch with whisper-tiny PLs. Note that adding supervised data significantly yields lower WERs.

models [188]. Yet, no consistent effect is noticed.

Regularization with supervised data Red graphs in Figure 4.3 also show WERs for offline models that along with PLs include a small amount of supervised data, up to 100h, for regularization. This strategy proves to be beneficial in cases with noisier PLs, particularly for smaller Whisper models like Whisper-tiny, Whisper-base, and Whisper-small when WER goes down for all the languages. The benefits, however, decrease or are absent with more accurate PLs generated by larger models, such as Whisper-medium and Whisper-large-v3. Thus, with our results on six languages, we can conclude that when supervised data is available, regularization is recommended for models with weak PLs and can be omitted with strong PLs. The results with 100h regularization are also available in Table 4.2 for offline models.

Scaling-up supervised data helps on cases with very noisy PLs For the ablation experiment on mixing in more supervised data, we maintain a fixed computational budget for generating PLs and explore the extent to which supervised data can offset noisy PLs. The results are pictured in Figure 4.4. Using only Whisper-tiny, we train TT models from scratch for non-English CommonVoice languages with over 200h of available supervised data (i.e., CA, DE, ES, FR, and IT). Our results show significant improvements in WER as supervised data increases from 100h to 200h and even more so up to 400h, especially in languages like Catalan, French, and Italian, which likely suffer from lower-quality PLs. For this experiment, our oracle results are from the models fully trained on the supervised data, which can be found in Table 4.2 for offline models.

Low-latency streaming decoding Figure 4.5 lists the streaming decoding results across six CommonVoice languages, testing 13 different decoding configurations ranging from 320 ms to 2560 ms chunks. We establish upper performance bound with models tested in non-streaming mode, and also include a box plot for each TT model trained with PLs derived from various Whisper model sizes. Larger Whisper models consistently yield lower mean WERs across all configurations, showing how model performance can fluctuate under different streaming conditions, with smaller chunk sizes or limited left context posing greater challenges.

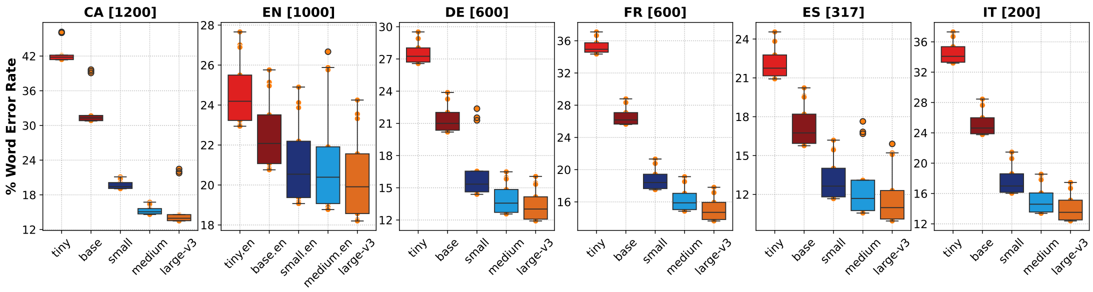


Figure 4.5: Box plots of WERs for six languages of CommonVoice. Streaming Zipformer models are trained from scratch, with only PLs generated with different Whisper model sizes. Each box denotes 13 decoding configurations, ranging from challenging (320ms chunk with limited left context) to more relaxed (2560ms chunk with full left context) streaming settings. (Note different WER scaling on the y-axis.)

Conclusions In this work, we address the challenge of training ASR systems with minimal supervised data by leveraging TT models trained from scratch using PL speech only derived from foundational speech models. We conduct a thorough examination of the efficacy of PL-based TT models across various dimensions, including offline and chunk-wise decoding for streaming applications, and the influence of FSM size on TT model's WERs. Our findings reveal that TT models can be effectively trained from scratch on noisy PLs, highlighting that this approach works even on low-quality PLs. We introduce robust heuristics to filter out unreliable and hallucinated PLs and explore the effects of FSM size on TT performance.

Table 4.2: WERs for six CommonVoice languages. The Zipformer models are trained with pseudo-labeled data from different Whisper models. We also report WERs when a small amount of supervised data is added during training, denoted as "sup. [100h]". Models are trained from scratch in ∼1 day GPU time and contain ∼70M parameters.

Experiment	**CA**	**EN**	**DE**	**FR**	**ES**	**IT**
	1200h	1000h	600h	600h	317h	200h
Whisper-tiny ($\theta = 39M$)						
Whisper model [188]	51.0	28.8	34.5	49.7	30.3	44.5
Zipformer + only PL	41.1	21.6	25.7	33.8	20.1	32.2
+ sup. [100h]	36.8	20.9	22.7	29.7	16.1	19.8
Whisper-base ($\theta = 74M$)						
Whisper model [188]	39.9	21.9	24.5	37.3	19.6	30.5
Zipformer + only PL	30.5	19.2	19.4	24.7	14.8	22.6
+ sup. [100h]	27.9	19.1	17.5	21.8	12.6	16.2
Whisper-small ($\theta = 244M$)						
Whisper model [188]	23.8	14.5	13.0	22.7	10.3	16.0
Zipformer + only PL	18.6	17.2	13.4	16.5	10.6	14.8
+ sup. [100h]	17.4	17.0	12.8	15.8	10.3	12.9
Whisper-medium ($\theta = 769M$)						
Whisper model [188]	16.4	11.2	8.5	16.0	6.9	9.4
Zipformer + only PL	14.0	16.7	11.3	13.7	9.5	12.1
+ sup. [100h]	13.7	16.5	11.3	13.5	9.5	12.0
Whisper-large-v3 ($\theta = 1.5B$)						
Whisper model [188]	14.1	9.4	6.4	13.9	5.6	7.1
only PL	12.8	16.2	10.6	12.4	8.9	11.1
+ sup. [100h]	12.8	16.3	10.7	12.3	9.1	11.6

4.2 Use of Large Pretrained Models for Transducer-based ASR

Self-supervised pretrained models exhibit competitive performance in automatic speech recognition on fine-tuning, even with limited in-domain supervised data for training. However, popular pretrained models are not suitable for streaming ASR because they are trained with full attention context. In this chapter, we introduce XLSR-Transducer, where the XLSR-53 model is used as encoder in transducer setup. Our experiments on the AMI dataset reveal that the XLSR-Transducer achieves 4% absolute WER improvement over Whisper large-v2 and 8% over a Zipformer transducer model trained from scratch. To enable streaming capabilities, we investigate different attention masking patterns in the self-attention computation of transformer layers within the XLSR-53 model.

Our contributions are covered below:

- Introduction of the XLSR-Transducer, a multilingual SSL encoder based transducer model, demonstrating significant WER improvement on the AMI dataset compared to large speech foundational models and other open-source ASR models;
- extension to streaming XLSR-Transducer and a systematic study of chunk size and past context on training and inference;
- to author's knowledge, this is the first work that explores the attention sink [221] phenomenon for streaming ASR which leads to improved WER; and
- Evaluation of the XLSR-Transducer on AMI and five languages of CommonVoice dataset in low resource settings.

Publication Note

The material presented in this section is adapted from the following publication:

- S. Kumar, S. Madikeri, J. Zuluaga-Gomez, E. Villatoro-Tello, I. Nigmatulina, P. Motlicek, M. K. E, and A. Ganapathiraju, "XLSR-Transducer: Streaming ASR for Self-Supervised Pretrained Models," in *arXiv:2407.04439*, 2024

Minor contributions Trained and validated the XLSR-Transducer models on the CommonVoice dataset. Experimental validation on the masking patterns to enable streaming decoding with XLSR-Transducer for CommonVoice dataset. Participated in the article write up.

4.2.1 Introduction

In streaming ASR, partial hypotheses are generated for each audio chunk sequentially [65, 220] to produce the transcript for the full audio, whereas the entire audio segment is available for non-streaming decoding. Depending on the latency requirements, the chunk size may vary from few hundred milliseconds to few seconds [220]. Typically, a drastic degradation in word error rate (WER) is observed when non-streaming models are decoded in streaming fashion [66], because only a limited context is available. In this work, we propose a variety of attention masking patters that enable streaming training and decoding of our XLSR-Transducer model. We also study the importance of chunk sizes and left context size by varying them during inference. For instance, at decoding time, increasing left context typically enhances ASR performance [220], at the expense of increased latency. Recently, it was shown that the transformer layers learn to assign disproportionate attention scores to few initial tokens for streaming language models [221], termed as attention sinks. We study the effects of attention sinks for the first time in streaming ASR. Formally, at decoding time, we allow the transformer layers in XLSR to attend to a few initial frames in addition to designated frames in chunk and past context. In theory, this reduces total computation required for processing an audio chunk during streaming decode.

4.2.2 XLSR-Transducer

In a typical Transformer-Transducer (TT) ASR model (Figure 4.6a), there are three networks: the encoder, predictor, and joiner. The encoder processes audio frames to produce acoustic embeddings. The predictor generates token embeddings in an auto-regressive manner, taking previous non-blank tokens as input. Lastly, the joiner combines the outputs from the encoder and predictor to predict a probability distribution over the tokens in the vocabulary. In this work, we utilize a stateless predictor [55] composed of an embedding layer and one 1-D CNN layer, and the joiner network consists of one linear layer. Typically, the encoders [56, 57, 58] are trained from scratch and require a large amount of in-domain supervised data to achieve decent WER, which may not always be feasible. We train the TT model using the pruned-transducer loss [60] from k2[11] toolkit.

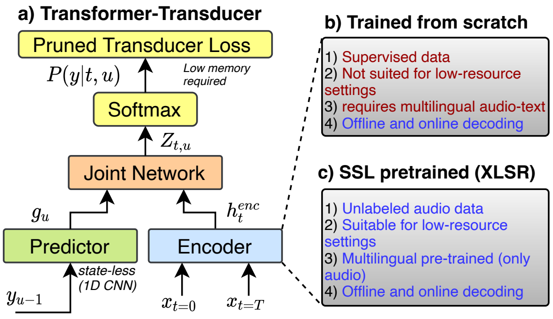


Figure 4.6: Current state-of-the-art a) Transducer ASR includes state-less predictor, pruned transducer loss and b) Transformer-based encoder, trained from scratch. We replace the encoder by XLSR-53, an SSL model suitable for low-resource applications. Our contributions lead to the c) XLSR-Transducer.

Non-Streaming XLSR-Transducer In contrast to encoders trained from scratch, recent advancements in SSL pretrained models demonstrate competitive performance [80, 51] when fine-tuned with a limited amount of labeled data for ASR. Previously, the ASR models employing SSL pretrained models have utilized CTC loss [45], encoder-decoder based architecture [70, 12], and Lattice-Free MMI loss [222] (hybrid approach) for training. Furthermore, we integrate pretrained models as encoders in the TT setup, as illustrated in Figure 4.6c. One notable advantage is the ability to achieve strong ASR performance with relatively low amounts of training data. We select XLSR-53 [51] as our encoder model, which takes raw audio as input and outputs audio frames with a frame duration of 25 ms and a stride of 20 ms. The selection of XLSR-53 is driven by its large-scale pre-training on multilingual audio, which has demonstrated competitive ASR

[11] https://github.com/k2-fsa/k2

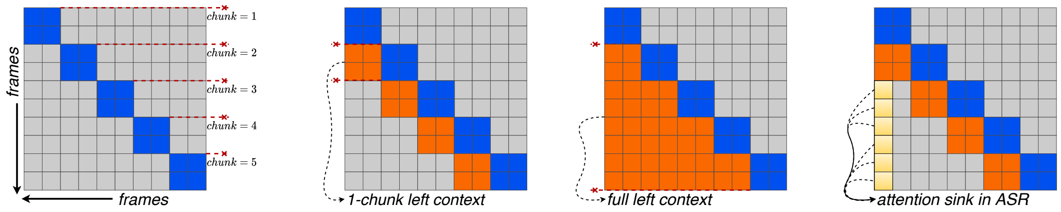


Figure 4.7: Masking strategies for streaming XLSR-Transducer. Multi-chunk training allows decoding with variable chunk size (blue) and left context (orange). Each square denotes “n” frames. Attention sink (yellow) allows context from the first n frames. Our results show that **attention sink frames offer a better trade-off w.r.t increasing left context alone**, leading to lower WERs.

performance [51] in the low-resource across multiple languages.

Streaming XLSR-Transducer XLSR is typically trained and decoded using the entire audio sample. This makes the proposed XLSR-Transducer non-streaming, despite the use of stateless predictor and linear joiner that are inherently streaming. The main challenge to port SSL models to streaming is the use of self-attention in the transformer layers [61], i.e., computed over entire acoustic frames of an utterance. Here, we present multiple masking patterns [220] to limit the frame context over which self-attention is computed, simulating streaming ASR within the XLSR-Transducer setup.

Chunked masking For a typical streaming ASR, decoding partial hypotheses should occur after receiving a few audio frames, known as the chunk size. As depicted in Figure 4.7a, we implement chunk-wise decoding by masking frames outside a specific chunk during the forward pass from the XLSR model. The mask is applied after dot-product computation during self-attention, ensuring that each frame inside a chunk has access to all the frames within that chunk. Note that the XLSR model also includes a CNN front-end, which takes raw audio as input. Thus, we feed chunk-size equivalent raw waves to the CNN front-end sequentially and concatenate them across the time dimension to obtain all the frames for an utterance. In this work, we explore chunk sizes of 16, 32, 64, and 128, translating to approximately 320 ms, 640 ms, 1280 ms, and 2560 ms, respectively, for XLSR.

Chunked masking with variable left context chunks In practice, when decoding chunk “n", we have access to all the previous chunks, which can be utilized as left context. As illustrated in Figures 4.7b and 4.7c, a variable number of left context chunks can be utilized during the self-attention computation of a chunk, with the possibility of using the full left context. The number of frames in the left context is a multiple of the chunk size, as this can be efficiently implemented to store past chunks in the cache.

Streaming training and decoding The use of non-streaming XLSR-Transducer for streaming decoding with the described masking patterns presents a challenge. The model has been trained on full context, creating a train-test mismatch. To address this challenge, we train the model in streaming fashion using a fixed chunk size and left context. Flexibility in our chunked mask implementation allow us to perform both streaming and non-streaming decoding using a single model. The advantage of our method is that it only affects the fine-tuning stage, and we can avoid the computationally prohibitive pre-training.

Multi-chunk Training In many practical use-cases of streaming ASR, varying the chunk size at decoding time is often desirable, depending upon latency requirements. However, a streaming XLSR-Transducer model trained with a fixed chunk size may not yield optimal WERs when decoded with different chunk sizes. Also, training multiple models for varying chunk sizes may be infeasible. To address this limitation, we propose randomly selecting the chunk size from the predefined list mentioned above (**chunked masking**) for each batch during training.

4.2.3 Efficient Streaming ASR with Attention Sinks

In a recent work on streaming language models (LM) [221, 223], it was shown that a surprisingly large amount of attention scores during self-attention computation inside transformer layers is directed towards the initial tokens, termed as *attention sinks*. This was attributed to the Softmax operation, which mandates attention scores to sum up to one and in autoregressive LMs, all subsequent tokens have access to the initial tokens. Consequently, the model may find it easier to learn to assign large scores to these initial tokens. In our streaming model training, where we utilize full left context, we employ a similar setup. This leads us to introduce the first utilization of the attention sinks in the context of streaming ASR during inference. Specifically, as depicted in Figure 4.7d, we enable self-attention to focus on not only frames within a chunk and left context chunks, but also on the initial few frames.

4.2.4 Databases and Experimental Setup

AMI and CommonVoice First, we train and evaluate the XLSR-Transducer model on the individual head microphone (IHM) split from the AMI dataset [110] containing audios with a sampling rate of 16 kHz. We use the default recipe for AMI from lhotse[12] toolkit to prepare the train, dev and eval sets containing 80h, 8.8h, 8.5h of audios respectively. In all our experiments on the AMI dataset, we use WER on the dev set to select the best epoch and report the results on the eval set. Second, we validate XLSR-Transducer on five non-English languages from CommonVoice-v11 [97].[13] This includes Catalan (CA), Belarusian (BE), Spanish (ES), French (FR) and Italian (IT). To keep experimentation under the low-resource domain, we extract randomly a 100h subset from the training data per language. Later, we train streaming and non-streaming models. We report WERs on the full official test sets.

[12] https://github.com/lhotse-speech/lhotse

[13] CommonVoice-v11: cv-corpus-11.0-2022-09-21)

Table 4.3: WERs on the AMI eval set. On non-streaming decoding† XLSR-Transducer yields significant WER reduction. On streaming decoding, the multi-chunk training (multiple) provides significant gain w.r.t encoders trained from scratch with minimal degradation in non-streaming performance. ‡encoder-decoder model. ¶decoding chunk size 2000 ms.

Encoder	**Chunk Size**	**Chunk Size decoding**		
	train-time	320 ms	1280 ms	full-att†
decoding: non-streaming ASR				
Whisper large-v2 (1.6B)‡ [188]	-	-	-	16.9
FastConformer (1.1B) [59]	-	-	-	15.6
Zipformer (70M)	-	-	-	21.0
decoding: streaming ASR				
FastConformer (114M)¶ [65]	-	-	24.2	-
Zipformer (70M)	multiple	28.5	24.6	23.2
XLSR (300M)	full-att	35.3	17.8	**12.7**
XLSR (300M)	320 ms	**17.1**	15.0	14.2
XLSR (300M)	1280 ms	19.7	14.5	13.1
XLSR (300M)	multiple	17.7	**14.2**	**12.9**

Zipformer-Transducer Baseline We establish strong baselines by training non-streaming and streaming Zipformer transducer models [58] from scratch, following the AMI recipe[14] from Icefall toolkit (we only use the IHM set). The (1) state-less Predictor [55] consists of an embedding layer and one 1-D CNN layer, (2) the joiner consists of 1 linear layer. We use the default hyperparameters and train for 30 epochs [58]. We use beam search with width of 4.

XLSR-Transducer Training The XLSR-transducer model is constructed from the Icefall's Transducer recipe for AMI dataset adapted with the XLSR model from fairseq [224]. The fine-tuning uses Scaled Adam [186] and a learning rate scheduler with a 500-step warmup phase [61] followed by a decay phase directed by number of steps and epochs. We optimize the model with pruned RNN-T loss [60, 53] with a learning rate of $lr = 1.25e^{-3}$ and $lr = 5.0e^{-3}$ for AMI and CommonVoice. We train AMI and CommonVoice models for 10 and 20 epochs, respectively.

4.2.5 Results & Discussion

Non-streaming ASR We benchmark first the XLSR-Transducer model for non-streaming ASR on the AMI dataset and the results are reported in the Table 4.3 (*full-attn*). We compare against large open source foundational speech models. It can be seen that the proposed XLSR-Transducer model achieves significant improvement in WERs. Specifically, it achieves a relative improvement of 19% in WER when compared to the best open source large foundational ASR

[14] github.com/k2-fsa/icefall/tree/master/egs/ami

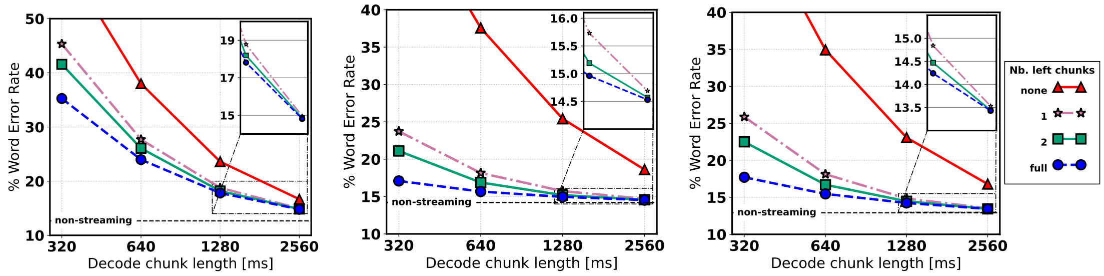


Figure 4.8: Non-streaming. Figure 4.9: Chunk-size of 16. Figure 4.10: Multi-chunk streaming.

Figure 4.11: Plots of WERs on AMI eval set for XLSR-Transducer trained on three configurations (a, b and c) and decoded on multiple streaming scenarios. Note that adding one or more left-context chunks at decoding time reduces WERs dramatically.

model. Next, we train a *Zipformer* encoder based transducer model from scratch and observe that XLSR-Transducer yields 39% relative improvement in WER. It is clear that there are significant advantages of using pretrained encoders in TT setup for low resource ASR.

Streaming ASR First, we decode the non-streaming trained XLSR-Transducer model in streaming fashion by applying different masks. Results are reported in the Table 4.3, where full left context is used during decoding. The XLSR-Transducer achieves significant improvement over *Zipformer* and *FastConformer* based transducer models. Despite the improvements, there is a significant degradation from non-streaming performance because the model was not trained for streaming. When the model was trained for streaming with full left context and decoded using a chunk size of 320 ms, the performance improves (35.3% → 17.1% WER) because of the train-test matched chunk size setting. We also train models with larger chunk sizes, but it degrades the performance, showcasing the importance of context during self attention computation inside a chunk which the model may have learned during training. As we increase the chunk duration during decoding (320 ms → 1280 ms), the performance improves monotonically [66], which is expected due to larger context available for frame attention score computation. Now, the streaming trained models are decoded in non-streaming, which can serve as performance upper bound. We observe that increasing chunk size during training improves the non-streaming performance and even when chunk size of 320 ms is used during training, the results only degrade by 1.5% in absolute WER. When random chunk sizes are used during training, the gap is 0.2% when compared with the best non-streaming ASR performance. Overall, a streaming trained XLSR-Transducer model using random chunk sizes shows best WER when decoded in streaming fashion with 1280 ms chunk duration and performance gap for non-streaming decoding is minimal. Thus, a single model can be used for both streaming and non-streaming ASR.

Streaming ASR with variable left context Using full left context during training and decoding will incur additional computation, which may not always be desirable. Limiting left context during training of streaming models resulted in significant degradation of results; therefore, we use full left context for all streaming XLSR-Transducer training. Figure 4.11 list WERs when

Table 4.4: WERs of streaming XLSR-Transducer on five CommonVoice languages. Models are fine-tuned on random 100h train subset and with multi-chunk training. full-att: non-streaming decoding. †CA is 28h long, and the remaining 26h. ‡median (duration) in seconds.¶non-streaming training and decoding.

Lang†	**Test Set**		**Streaming model - chunk size [ms]**					**full-att**¶
	#utt	[50%-dur]‡	320	640	1280	2560	full-att	full-att
CA	16k	6.1	17.5	15.2	13.9	12.9	12.0	10.7
BE	15.8k	5.7	20.0	17.5	15.9	14.8	13.8	13.7
ES	15.5k	6.1	17.7	15.0	13.5	12.2	11.3	10.8
FR	16k	5.7	24.3	21.6	20.0	18.7	17.6	17.1
IT	15k	6.3	18.5	15.9	14.3	13.1	12.1	11.5

the number of left context chunks is varied during inference. Note that the left context duration is in multiple chunk size. Increasing the left context improves the performance for all training scenarios and chunk duration. At the same time, a significant improvement is observed using just one chunk of left context. Further increase in left context improves WER overall, but the benefit per left context chunk is lower. When the XLSR-Transducer is trained with multi-chunk streaming strategy and decoded with 1280 ms chunk size, a relative improvement of only 4% in WER is observed using full left context instead of one left context chunk. Thus, a limited number of left context chunks should be enough for most real-world streaming ASR.

XLSR-Transducer on multiple languages We also train the proposed model on five non-English languages of CommonVoice [97]. WERs are listed in Table 4.4 for multi-chunk streaming and full-attention non-streaming models. We see competitive WERs for models evaluated under different streaming conditions, with constant WERs improvement as chunk size increases; similar behavior is reported in [66]. The upper-bound WERs are obtained with a model trained and evaluated in non-streaming fashion, i.e., last column of Table 4.4. We note negligible WER degradation (up to 1.5% absolute WER, worse CA; best BE) for full-attention decoding (*full-att*) on streaming models vs. their non-streaming counterparts. This confirms the robustness of XLSR-Transducer on multiple languages.

Streaming ASR with attention sinks In theory, restricting left context chunks should lead to overall latency improvements for streaming ASR. The recent observation of attention sinks phenomena [221, 225], where the transformer models learn to assign relatively higher attention scores to initial tokens, may help in reducing the overall computation required to decode one chunk of audio in streaming ASR. WERs on the AMI dataset are reported in Table 4.5. For different chunk sizes and left context, we observe that increased frames for attention sinks improve the performance monotonically. Specifically, for a smaller chunk of 320 ms, using 1 left context chunk and 16 frames (i.e., 320 ms) for attention sinks performs better than using 2 left context chunks by 12% in relative terms despite attending over the same number of frames. We do not observe a significant reduction in WERs beyond a chunk size of 640 ms. Overall, our

Table 4.5: WERs on AMI eval set for varied decoding settings. Adding attention sinks offer a better trade off than larger left context. (blue) denotes relative WER reduction w.r.t no attention sink within same chunk and left context. †nb. of chunks. ‡nb. attention sink frames.

Decoding settings		**Decoding chunk-size**	
Left-context†	**attn-sink‡**	**320 ms**	**640 ms**
full	none	17.7	15.5
1	none	25.9 –	18.1 –
	1	22.9 (+11.6)	17.4 (+4.1)
	4	21.4 (+17.4)	16.8 (+7.1)
	16	19.7 (+23.7)	16.3 (+10.0)
2	none	22.5 –	16.7 –
	1	20.9 (+7.3)	16.4 (+1.7)
	4	20.0 (+11.3)	16.2 (+3.2)
	16	18.9 (+16.1)	15.9 (+4.8)
4	none	19.8	15.9

results show that it is better to use attention sinks than increasing left context chunks beyond 1 for improved performance. We also run decoding with attention sinks on the CommonVoice dataset and observe similar trends but do not include a results table for brevity.

Conclusions This section demonstrates that using an SSL pretrained model as encoder in the transducer framework, termed as XLSR-Transducer, leads to significant improvement in WER on AMI and multiple languages from CommonVoice corpora. We explore various chunked masks and left context configurations to enable streaming decoding in XLSR. Our findings across 2 datasets and 6 languages shows that the proposed model achieves streaming performance comparable to non-streaming ASR.

4.3 Improved Streaming Transducer With Attention Sinks

Chunk-wise decoding with left context is a popular choice for streaming automatic speech recognition (ASR). In the most challenging applications, limited left context is enforced to reduce processing time and computational budget. In this work, we explore novel phenomena from the NLP domain called *attention sink*, where at decoding time, we allow attention to initial frames in addition to left context chunks. We validate this on Transformer-Transducers (TT) models trained from scratch for more than 10 languages of CommonVoice under several low-resource ASR settings, ranging from 17h to 100h of fine-tuning supervised data. We show that on challenging streaming settings with limited left context history, attention sink yields a 12% word error rate (WER) reduction w.r.t increasing left context alone, thus being more computational friendly. This is an extension of Section 4.2.

Our contributions are covered below:

- we extend the XLSR-Transducer model [11] by an exhaustive ablation of the attention sink phenomena within the streaming TT ASR framework,
- study of attention sink on more than 10 languages from CV, where we show consistent WERs reduction with regard to only increasing left context alone;
- we ablate XLSR-Transducer equipped with attention sink on low-latency settings with chunks from 320 ms up to 2560 ms.

Publication Note

The material presented in this section is adapted from the following publication:

- J. Zuluaga-Gomez, S. Kumar, *et al.*, “Improved Streaming Transformer Transducer With Attention Sinks,” in *To be Submitted to ARR (long paper)*, 2024

Minor contributions This is an extension of Section 4.2. I trained and validated the XLSR-Transducer models on CommonVoice for attention sink phenomena on multiple languages. Lead the work, including the paper write up.

4.3.1 Introduction

Automatic speech recognition has been largely impacted by end-to-end (E2E) modeling to an extent that it has become ubiquitous in the literature. Currently, the most prominent E2E architectures includes Connectionist Temporal Classification (CTC) [45, 80, 51], attention-based encoder-decoder (AED) [188, 70], and the neural Transducer architectures [53, 60]. While CTC and AED-based models are non-streaming by design, they can be ported to streaming at inference time, albeit with caveats e.g., drastic downstream WER degradation or hallucinated outputs [226, 227]. However, neural transducers–or Transformer-Transducer (TT)–support streaming decoding by nature, thus we focus from now only on this architecture. In streaming ASR, we generate partial hypotheses sequentially for incoming audio. The audio is processed

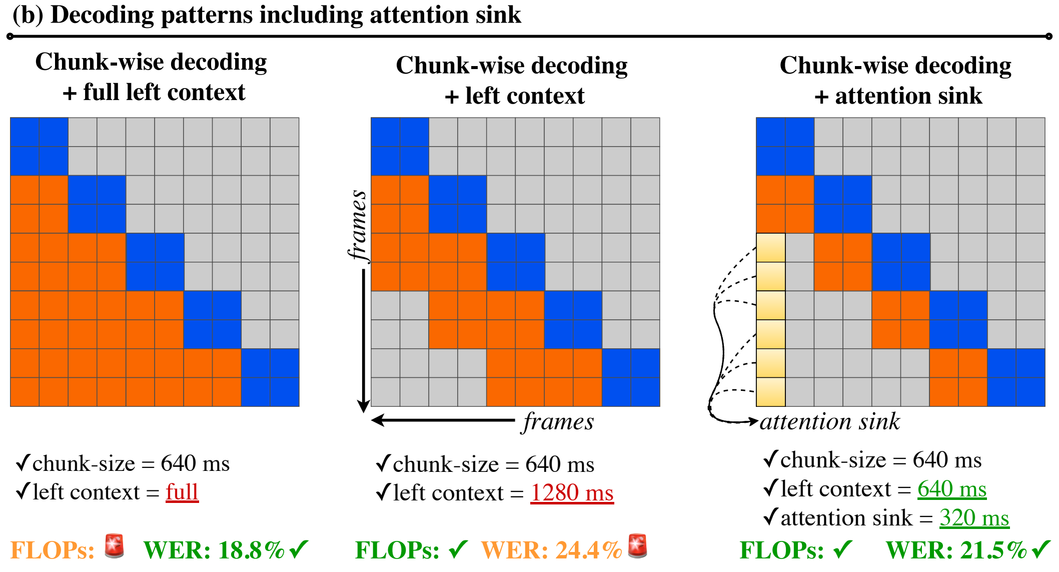


Figure 4.12: The attention sink effect when decoding with limited left context. It is more efficient than standard decoding and yields ∼12% WER reduction w.r.t increasing left context alone. WERs for models trained with 100h of data and averaged across 13 languages of CommonVoice.

in a chunk-wise manner until we reach the end of the stream [220]. Aspects such as chunk size, left context and its size can determine the quality of the hypothesis and the latency of the system. This contrasts with non-streaming decoding methods, where the entire audio segment is available for processing at once, i.e., full bidirectional attention decoding.

Despite this rapid growth of interest, there are some caveats of E2E models: (1) AEDs do not support streaming by design and (2) transducers models rely on large-scale supervised databases. Authors in [11] propose to warm start the encoders in TT models with the XLSR-53 model [51]. This bridges the shortcoming of each architecture, while showing competitive WERs on the low-resource streaming ASR settings. Furthermore, the authors also explore interesting phenomena seen in large language models, *attention sink* [221]. Here, we do an extensive study on of these aspects and show how to improve model performance under challenging low-latency streaming settings.

4.3.2 Related Work

There are multiple approaches to improve WERs and latency for AED and transducer models, including (1) faster word emition with FastEmit [228], or self alignment [229]; (2) model sparsification [202], quantization [54] and pruning [230, 231]; (3) efficient transducer loss functions, e.g., pruned RNN-T [60], or (4) efficient encoders, e.g., HyperConformer [12], FastConformer [59, 65] or stochastic compute reduction for wav2vec 2.0 [232]. In this work, we focus on improving WERs at decoding time under challenging streaming scenarios, e.g., chunk-wise decoding with limited chunks and left context.

4.3.3 Databases and Experimental Setup

CommonVoice and TEDLIUM Databases We use CommonVoice database for experimentation. See Table 2.1 for further details about each language split. Similarly, we use TEDLIUM for long-form ASR, see further information about this dataset in Section 2.3.1.

XLSR-Transducer Training and Decoding The XLSR-transducer model is constructed from the Icefall's Transducer recipe for AMI dataset [110], adapted with the XLSR model from fairseq [224]. We follow the same implementation as in [11], see Section 4.2 for further details. The model is optimized with pruned RNN-T loss [60, 53]. For experiments with CommonVoice we train for 20 epochs on languages with 100h and for 50 epochs for the ones below that. For TEDLIUM-v3 we train for 10 epochs. **For decoding**, we use multiple masking patterns, similar as in [11]. This includes chunk-wise decoding with (1) full left context, (2) limited left context, and (3) limited left context with attention sink. As described in Figure 4.12.

Attention sink experiments We run several ablations to validate the attention sink pattern. We select 10 languages from CommonVoice and train XLSR-Transducer with up to 100h of labeled data. This ensures that our models are under the mid-to-low resource setting.[15] Five languages are under the low-resource setting, all the way to 17h of labeled data (CS).

4.3.4 Results & Discussion

We run experiments for each CommonVoice language by training streaming XLSR-Transducer models in a multi-chunk fashion, i.e., we use masking during training with different configurations, see [11, 220]. More information in Section 4.1.

Non-streaming decoding In Table 4.6, we present the baseline WERs for the XLSR-Transducer evaluated in a non-streaming mode, serving as an upper performance bound. We also compare these results with strong AED models such as Whisper [188], despite Whisper models not being trained on CommonVoice but on substantially larger datasets. Our findings show that XLSR-Transducer WERs are competitive with Whisper models of similar size, and occasionally even with the larger Whisper-large-v2. We also run a very low-resource experiment with Swedish (sv-SE), where we only use 7h of supervised data.

Streaming decoding with attention sink In low-latency settings, reducing left context can attenuate computational demands, potentially degrading WERs. The integration of attention sink significantly counters this degradation by enhancing ASR performance even under exigent

[15] Note that we do not aim to get the best WER with our models, as a standard training from scratch with the full training corpus most surely lead to lower WERs.

Table 4.6: Full WERs on 14 CommonVoice languages and comparison w.r.t multiple Whisper models. †only available for whisper-large-v3. ‡substantial improvements are seen when fine-tuned to 1kh instead of 100h.

Supervised data [h.]	CA 100h	BE 100h	ES 100h	DE 100h	FR 100h	IT 100h	EN 100h	SW 100h	RW 100h	NL 33h	PL 24h	PT 20h	RU 32h	CS 17h	sv-SE 7h
Baselines from Whisper [188].															
small ($\theta = 244M$)	23.8	-	10.3	13.0	22.7	16.0	14.5	-	–	14.2	16.9	12.5	15.0	34.1	22.1
medium ($\theta = 769M$)	16.4	-	6.9	8.5	16.0	9.4	11.2	-	–	8.0	10.1	8.1	9.3	18.8	13.7
large-v2 ($\theta = 1.5B$)	14.1	43.7†	5.6	6.4	13.9	7.1	9.4	51.2†	–	5.8	7.6	6.3	7.1	13.5	10.6
XLSR-Transducer model															
XLSR-T ($\theta = 317M$)	10.7	13.1	10.2	13.4	16.4	10.7	20.3	18.5	35.5‡	14.0	17.6	13.5	21.2	28.2	43.6

conditions (320 ms chunks without lookahead or substantial left context), as shown in Figure 4.13. Across all tested languages from CommonVoice, adding 1 to 16 frames of attention sink consistently lowers the WER. Note that these experiments are very challenging as (1) the chunk size is low, i.e., 320 ms, (2) we do not use any look-head or future context, and (3) the model needs to work with very small left context.

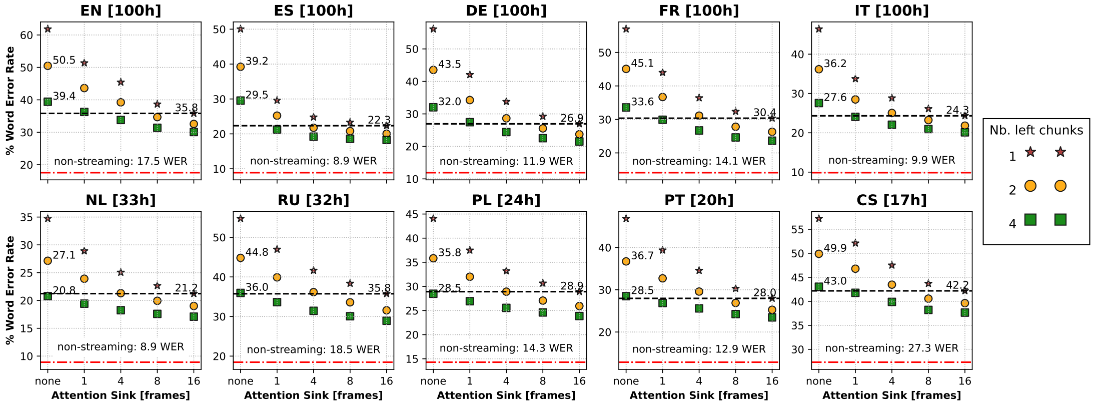


Figure 4.13: WERs per language for low latency chunk-wise decoding, chunk=320 ms. X-axis: number of attention sink frames. Top row: high-resource languages with 100h of training data; bottom row: mid-to-low resource setting, below 35h. Red line: non-streaming decoding. Dark line: WER for the experiment with a single left context chunk and 16 attention sink frames.

Low-resource ASR with XLSR-Transducer Further insights from Figure 4.13 reveal that even with up to 33h of supervised data, the use of attention sink yields notable WER improvements in languages like NL, RU, PL, and PT. This shows the robustness of XLSR-Transducer equipped with attention sink for low-resource and challenging streaming conditions.

Attention sink on multiple streaming settings Comprehensive results for 13 CommonVoice languages under varying conditions are detailed in Figure 4.14. We assess multiple chunk sizes (16/32/64, i.e., 320 ms/640 ms/1280 ms) and explore the impact of introducing 1 to 16 frames of

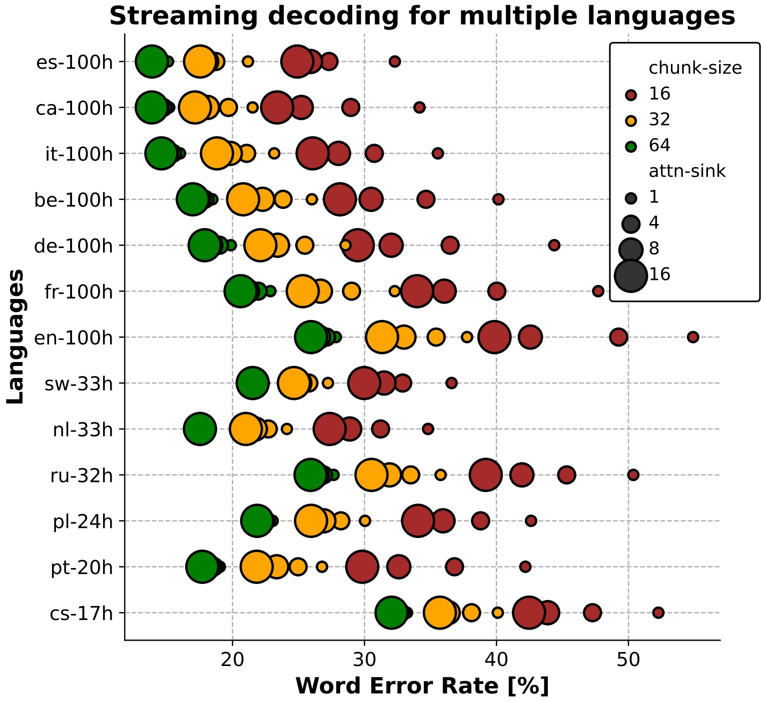


Figure 4.14: Multiple decoding results per language, including attention sink frames. Decoding is reported only with left context chunks of 1.

attention sink, offering a broad view of XLSR-Transducer performance across different streaming configurations.

Attention sink on TEDLIUM Table 4.7 list the complete results under different decoding settings for XLSR-Transducer models trained on TEDLIUM. We also list the upper-bound WER of 7.2% WER for a system trained and decoded in non-streaming fashion. As shown in Table 4.7, applying attention sink in configurations with two or fewer left context chunks significantly benefits the WER across all settings. This establishes the value of attention sink in enhancing long-form ASR.

Scaling-up the training data We observe drastic WER reductions (as shown in Table 4.8) by substantially increasing the training data volume for six languages from CommonVoice. In several instances, our models surpass the performance of Whisper-large-v2, particularly in languages like CA, FR, and BE, illustrating the scalability and effectiveness of the XLSR-Transducer with larger training sets.

Conclusions In conclusion, this study validates the phenomena of “attention sinks" within the streaming Transformer-Transducer framework for ASR. We demonstrate that a significant

Table 4.7: Complete streaming decoding with attention sink ablation for TEDLIUM dataset. †number of left chunks at decoding time, which depends on the chunk size per experiment.

#LC†	**cs=320 ms**					**cs=640 ms**					**cs=1280 ms**					**cs=2560 ms**				
chunk	0	1	4	8	16	0	1	4	8	16	0	1	4	8	16	0	1	4	8	16
0	85.8	69.0	62.0	60.0	57.5	52.6	43.4	40.0	38.9	37.7	27.0	25.4	23.8	23.2	22.3	16.2	15.6	14.9	14.6	14.3
1	33.5	21.7	15.2	14.2	13.2	16.7	14.2	11.8	11.3	10.9	10.9	10.4	9.6	9.4	9.3					
2	24.2	18.2	13.9	12.9	12.3	13.8	12.6	11.0	10.7	10.5	9.7	9.5	9.1	9.1	9.0					
4	17.5	15.1	12.7	12.2	11.9	11.5	11.1	10.4	10.2	10.0	9.0	9.0	8.9	8.9	8.8					
8	13.2	12.6	11.8	11.5	11.3	10.2	10.2	10.0	10.0	10.0	8.7	8.7	8.7	8.7	8.7					
full	10.9					9.8					8.7					8.2				
	non-streaming training and decoding: 7.2% WER																			

Table 4.8: Impact of scaling up the training data in XLSR-Transducer. We train XLSR-Transducer models with larger train subsets and report WERs with beam search decoding and model averaging of 5. †only available for whisper-large-v3.

	CA	EN	RW	DE	FR	BE	ES	IT
Baselines from Whisper [188].								
whisper-medium ($\theta = 769M$)	16.4	11.2	-	8.5	16.0	-	6.9	9.4
whisper-large-v2 ($\theta = 1.5B$)	14.1	9.4	-	6.4	13.9	43.7†	5.6	7.1
XLSR-Transducer model ($\theta = 317M$)								
W/ 100h sup. data	10.7	20.3	35.5	13.4	16.4	13.1	10.2	10.7
W/ scaled-up: +sup. data	1kh	1kh	1kh	600h	600h	400h	300h	200h
	5.8	14.5	21.8	9.0	11.4	7.1	7.7	8.8

improvement in WERs across multiple low-resource languages from the CommonVoice dataset are achieved under low-latency settings with limited left context. By allowing attention to focus not only on the immediate left context but also on initial frames, our approach effectively reduces computational demands while enhancing WERs. The results presented in this section reaffirm our earlier results presented in [11] and further discussed in Section 4.2.

4.4 Compute-Bounded Low-Resource Speech Recognition with HyperConformer

State-of-the-art ASR systems have achieved promising results by modeling local and global interactions separately. While the former can be computed efficiently, global interactions are usually modeled via attention mechanisms, which are expensive for long input sequences. Here, we address this by extending HyperMixer [233], an efficient alternative to attention exhibiting linear complexity, to the Conformer architecture for speech recognition, leading to HyperConformer.

Our contributions are covered below:

- Multi-head HyperConformer achieves comparable or higher recognition performance while being more efficient than Conformer in terms of inference speed, memory, parameter count, and available training data;
- HyperConformer achieves a word error rate of 2.9% on LibriSpeech test-clean with less than 8M neural parameters and a peak memory during training of 5.7GB, hence trainable with accessible hardware; and
- encoder speed is between 38% on mid-length speech and 56% on long speech, faster than an equivalent Conformer.

Publication Note

The material presented in this section is adapted from the following publication (shared first authorship):

- F. Mai, J. Zuluaga-Gomez, T. Parcollet, and P. Motlicek, “HyperConformer: Multi-head HyperMixer for Efficient Speech Recognition,” in *Proc. Interspeech*, 2023, pp. 2213–2217

Supplementary materials related to this section:

- **Open-source HyperConformer model on SpeechBrain GitHub:** https://github.com/speechbrain/speechbrain/blob/develop/speechbrain/nnet/hypermixing.py
- **Recipe for ASR training on SpeechBrain GitHub:** https://github.com/speechbrain/speechbrain/blob/develop/recipes/LibriSpeech/ASR/transformer/hparams/hyperconformer_22M.yaml

Major contributions Problem definition and experimental design and setup. Data preparation. Trained the encoder-decoder models for the experiments. Co-lead the work, including the integration of HyperMixer in the speech encoder and the paper write up.

4.4.1 Introduction

Automatic Speech Recognition (ASR) technologies have greatly benefited from deep learning, reaching unprecedented levels of accuracy and pushing successful products to real-life use cases. Various architectures of ASR systems co-exist and deliver superlative performance depending on the task or domain of interest [234]. A prevalent family of ASR systems uses self-attention and Transformer neural networks to consume the input speech sequence and build powerful representations both at the acoustic and linguistic levels [235]. Indeed, the ability of Multi-Head Self-Attention (MHSA) [61] to capture long-term dependencies via its sequence-long receptive

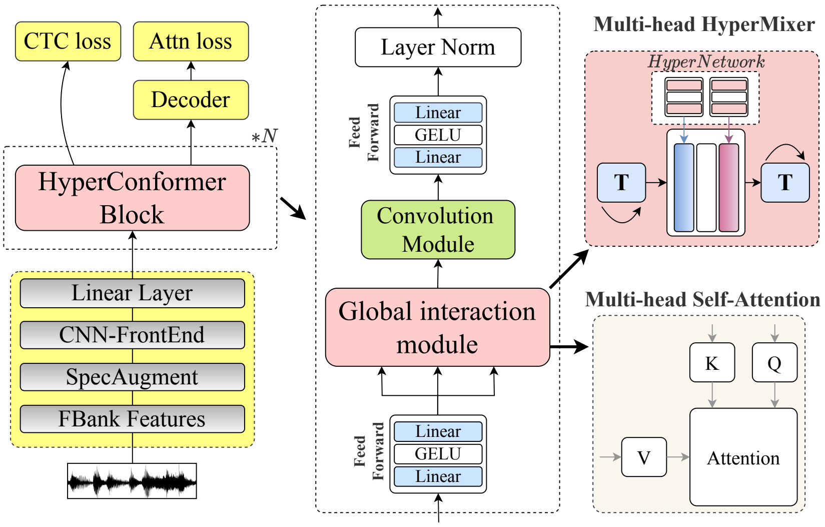


Figure 4.15: Layout of the general Conformer architecture. Global interactions can be modeled either with attention leading to a *Conformer* or with HyperMixer to obtain *HyperConformer*. **T** represents the transpose operation. Skip connections are omitted for simplicity. The global interaction module is combined sequentially with a convolution module to capture local dependencies, critical for speech-related tasks.

field helped Transformer ASR architectures to outperform the previous state-of-the-art mostly composed with recurrent neural networks [235]. Nevertheless, ASR not only requires capturing global interactions describing the semantic and linguistic characteristics of the speech utterance, but also modeling properly the local interactions that form the speech signal.

Conformer neural networks [57] have been introduced to specifically address this issue. They combine Transformer and Convolutional Neural Network (CNN) blocks to capture the global and local dependencies, respectively, leading to improved Word Error Rate (WER). Most prominently, variations of the Conformer, named Branchformer [236] and E-Branchformer [237] reached the lowest WER on the widely-adopted LibriSpeech dataset [94] while being trained from scratch without external data. Following the local and global dependencies' assumption, Branchformer architecture physically create two branches per block (a dual path) in the architecture to capture independently and with adapted mechanisms (i.e., MHSA and CNN) both levels of dependencies. The latter branches are then merged and passed to the next architecture block. Such approaches are agnostic to the type of ASR decoding or processing, e.g., Transducers [53], CTC only [45], or CTC and attention [70]. However, they suffer from a major and well-documented efficiency issue, as MHSA exhibits a quadratic complexity and memory time-dependency [233]. For instance, the MHSA block is among the most computationally demanding elements of any Transformer model. This is especially true for speech processing, as input sequences are often long by nature, e.g., longer than 30 seconds for a few LibriSpeech utterances [238, 239]. In addition, large-scale and Transformer-based Self-Supervised Learning (SSL) models for speech recognition are commonly trained with sentences voluntarily cropped at 20 to 25 seconds. The latter transformation is necessary to enable training with top-tier GPU e.g., Tesla V100 or A100 [50], also making it potentially intractable to train on more accessible compute infrastructures. This work focuses on retaining MHSA's global interactions capabilities beneficial to ASR while lowering significantly its computational and memory cost.

How to efficiently compute interactions between tokens in Transformer-like architectures is an active area of research [240]. Most works try to decrease the cost of attention directly, e.g., through a low-rank approximation [241], linearization [242], clustering [243], or the introduction of sparse attention patterns [244]. However, token mixing can also be achieved from outside the framework of attention, opening up considerably novel opportunities for improvement. MLPMixer [245] was the first to learn a fixed-size MLP for modeling global interactions, with many to follow in the vision domain [246, 247, 248]. However, the fixed size hinders their adoption for domains with variable length signals. Existing approaches for speech have strong locality biases [249, 250] and still rely on small attention modules for the best performance [250]. Recently, [233] proposed HyperMixer for text processing, which achieved competitive performance to attention at a substantially lower cost in terms of computation and data. Intuitively, HyperMixer constructs the token-mixing MLP of MLPMixer *dynamically as a function of the data*, hence being amenable to variable length inputs.

4.4.2 HyperMixer Architecture

Figure 4.15 illustrates the different blocks of the introduced HyperConformer. It consists of four parts: Two feature mixing layers (feed-forward networks) at the bottom and top of the layer, a module for modeling local interactions, specifically the CNN introduced in [57], and a global interaction module. In the following, we discuss the global interaction modules bringing token mixing to the model. Other components of HyperConformer are identical to the Conformer [57].

Capturing Global Interactions Let $X \in \mathbb{R}^{N \times d}$ represent N d-dimensional token vectors, also equivalent to a latent representation of speech coming from the previous layer on length N. The global interaction module $\mathrm{GI} : \mathbb{R}^{N \times d} \rightarrow \mathbb{R}^{N \times d}; X \mapsto X'$ is responsible for combining information from different tokens in such a way that every $X'_{:,j}$ contains information from every $X_{:,i}$. Such a behavior captures global interactions as it interconnects the different time steps of the given speech or latent sequence. This may be achieved, for instance, via multi-head attention or via HyperMixer.

Multi-Head Self-Attention At the core, Multi-Head Self-Attention (MHSA) [61] relies on scaled dot-product attention:

$$\text{Attention}(X) = \text{Softmax}(\frac{XX^T}{\sqrt{d_k}})X,$$

which involves computing the dot product between every pair of input tokens, invoking memory and runtime complexity of $\mathcal{O}(N^2 \cdot d)$. The latter is responsible for the quadratic increase in memory and time consumption of standard Transformer architectures [233]. Further modeling capabilities are commonly obtained with the introduction of k parallel heads, allowing the model to attend to information from different representation subspaces, i.e., different views of the data:

$$\begin{aligned}\text{MHSA}(X) &= \text{Concat}(\text{head}_1, \ldots, \text{head}_k)W^O, \\ \text{head}_i &= \text{Attention}(XW_i^Q, XW_i^K, XW_i^V),\end{aligned}$$

with W^O, W_i^Q, W_i^K, W_i^V learnable weight parameters.

HyperMixer From a high-level perspective, HyperMixer achieves token mixing over variable length sequences by dynamically constructing a *token mixing MLP* through the use of hypernetworks [251]. The latter models specialize in generating neural network parameters, e.g., weights and biases. A token-mixing MLP is a multilayer perceptron $\text{TM-MLP} : \mathbb{R}^{d \times N} \rightarrow \mathbb{R}^{d \times N}$ that combines information from different tokens *for each feature independently*, e.g., processing the Fbank coefficients of each time step of a sequence:

$$\text{TM-MLP}(X)_{i,:} = \text{LayerNorm}(W_1(\sigma(W_2^T X_{i,:}^T))), \tag{4.2}$$

where $W_1, W_2 \in \mathbb{R}^{N \times d'}$ are weight matrices with the hidden layer size d'. σ represents some non-linear activation function; we fix it to GELU [158] following [233]. Furthermore, we add layer normalization [252] for improved stability. Intuitively, the input layer W_1 decides to what degree each token's information should be sent to the hidden layer of TM-MLP, and the output layer W_2 decides for each token what information to extract from the hidden layer.

Importantly, W_1, W_2 themselves are not learnable parameters, which would require the input to be of the same fixed size at all times. Instead, $\text{HyperMixer}(X; d, d')$, parameterized through the embedding dimension d and the hidden layer size d', first dynamically generates W_1, W_2 from the inputs themselves with the two hypernetworks $\text{MLP}^1, \text{MLP}^2$:

$$W_k(X) = \begin{pmatrix} \text{MLP}^{\text{k}}(X_{:,1} + p_{:,1}) \\ \vdots \\ \text{MLP}^{\text{k}}(X_{:,N} + p_{:,N}) \end{pmatrix} \in \mathbb{R}^{N \times d'}, k \in 1, 2.$$

$\text{MLP}^1, \text{MLP}^2 : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ contain the learnable parameters of HyperMixer, and $p_{:,j}$ are absolute position embeddings from standards Transformers [61]. After generating the weights, Equation 4.2 is applied. This determines the complexity of this model: $\mathcal{O}(N \cdot d \cdot d')$, which is the same asymptotic runtime as the feature mixing layers. Hence, HyperMixer turns the quadratic memory and inference time complexities to a linear regime.

4.4.3 Multi-head HyperMixer for Efficient ASR

Analogously to MHSA, we propose an extension of HyperMixer to multi-head HyperMixer (MHHM) and HyperConformer, by introducing multiple token mixing heads. To this end, we create k parallel $\text{HyperMixer}^l(\cdot; d/k, d'/k), l \in 0..k-1$, which each operates on (d/k)-dimensional feature subsets of X, whose outputs are again concatenated:

$$\begin{aligned} \text{head}_l &= \text{HyperMixer}^l(X_{:,(l\cdot(d/k)):(l+1\cdot(d/k))}) \\ \text{MHHM}(X) &= \text{Concat}(\text{head}_1, \ldots, \text{head}_k) \end{aligned}$$

As a result, and conversely to MHSA, the runtime complexity even further reduces to: $\mathcal{O}(k \cdot (N \cdot (d/k) \cdot (d'/k))) = \mathcal{O}(\frac{N \cdot d \cdot d'}{k})$.

4.4.4 Experimental Setup

Our experiments aim at assessing the effectiveness and efficiency of HyperConformer in comparison to Conformer. Hence, we compare vanilla Transformer [61] and Conformer [57] models to HyperMixer and HyperConformer. In practice, we swap the global interaction module, i.e., attention, from `regularMHA` (which uses absolute position embeddings [61]) of Transformer and `RelPosMHAXL` (which uses relative position embeddings [253]) of Conformer to our multi-head `HyperMixer` implementation.

Datasets and decoding We validate HyperConformer, on the LibriSpeech dataset [94]. It is composed of ∼960h of transcribed speech in English. We perform ablations either training on the 100h set or the full, 960h set, and report results on the dev/test sets and clean/other partitions. Additionally, we use the text-only corpus[16] for external language modeling (LM).[17] The LM is a Transformer based [61] only-encoder model composed of 12 encoder layers, $d_{ffn} = 3072$ and $d_{model} = 768$, which accounts for 93.3M parameters. Word error rates are reported using beam search with and without LM shallow fusion.

Neural architectures To gain a comprehensive understanding of performance and primary trade-offs, we ablate four different architectures in an encoder-decoder style: i) vanilla Transformer, ii) Conformer, iii) HyperMixer, and iv) HyperConformer. For the efficiency analysis only, we also experiment with replacing `RelPosMHAXL` with `regularMHA` (Conformer-regular). All models use a 5K BPE sub-word unit [67] vocabulary. This remains consistent across all experiments and models. At the bottom of the encoder, we incorporated a front-end module consisting of a 2-layer CNN that receives 80-dim log Mel filterbank features. We use SpecAugment [160]

[16] See https://www.openslr.org/resources/11/librispeech-lm-norm.txt.gz.

[17] Pretrained LM from SpeechBrain available in:
huggingface.co/speechbrain/asr-conformersmall-transformerlm-librispeech.

Table 4.9: WERs on the official LibriSpeech dev and test sets for models trained on the 960h LibriSpeech set. The results include the four proposed encoder models, including our novel architecture, HyperConformer. We ablate two different model sizes for each architecture and list results with and without LM. The last column list the peak memory consumption [GB] of each architecture under the same training conditions.

Model	**Par.**	**WER w/o LM**				**WER w/ LM**		**Peak**
		dev		**test**		**test**		**Mem.**
	[M]	clean	other	clean	other	clean	other	*[GB]*
Small sized models ($d_{model} = 144$)								
Transformer	6.1	7.7	15.6	7.8	15.8	3.9	8.2	6.45
HyperMixer	5.6	12.9	23.1	13.1	23.4	5.8	12.6	4.04
Conformer	8.7	4.7	11.4	5.0	11.3	3.1	6.8	8.18
HyperConformer	7.9	5.0	12.1	5.3	12.3	2.9	7.0	5.67
Medium-sized models ($d_{model} = 256$)								
Transformer	16.2	4.6	10.7	4.7	10.9	2.7	6.1	7.6
HyperMixer	14.4	7.2	15.2	7.5	15.2	3.9	8.3	5.6
Conformer	24.1	3.6	8.8	3.8	8.7	2.6	5.9	10.7
HyperConformer	21.7	3.4	9.0	3.6	9.0	2.3	5.7	8.6

during training with the default configuration in SpeechBrain. To correspond to accessible hardware as well as to emphasize low-compute resources performance, all models are conceived within a 25M parameter budget and trained with an 11GB memory constraint, corresponding to accessible GPU such as the Ti 80 family (or Ti 70 for the small version of HyperConformer). Hence, we select two model sizes for each architecture, i.e., 8 different scenarios. We use the same configuration, 10 encoder layers, and 8 attention or HyperMixing heads. However, we set $d_{model} = \{144, 256\}$ for `{base, medium}` models, respectively. The feed-forward network dimensions is set to $d_{ffn} = 4 \cdot d_{model}$ for all cases. For simplicity, we set the hidden layer size d' of TM-MLP to $d' = d_{ffn}$. We leave an exploration of this hyperparameter to future work.

Training hyperparameters Training is performed by combining the per-frame transformer decoder output probabilities and CTC [235]. The CTC loss [45] is weighted by $\alpha = 0.3$ during training. All the models use the same decoder, i.e., 4 Transformer layers. We follow the default training configuration of the LibriSpeech recipe from SpeechBrain.[18] It uses Adam [186] optimizer, learning rate ($lr = 1e^{-3}$) scheduler with warmup [61] (25k steps warmup). We train for 110 epochs, i.e., ∼660k steps when full LibriSpeech and ∼70k when LibriSpeech 100h set. The recipe also uses dynamic batching, which reduces the overall training time. At decoding time, we use a beam size of 66 with a CTC weight of $ctc_w = 0.4$. All of our experiments can be run on accessible GPUs starting from the Ti 70 family.

[18] Please refer to the SpeechBrain recipe located in `recipes/LibriSpeech/ASR/transformer`.

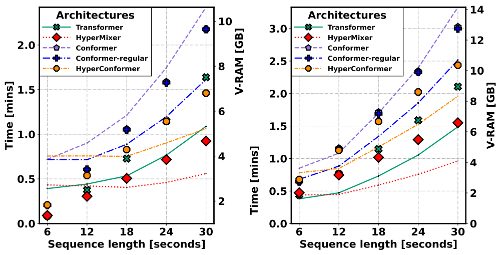


Figure 4.16: Forward pass of small (left) and medium sized (right) models.

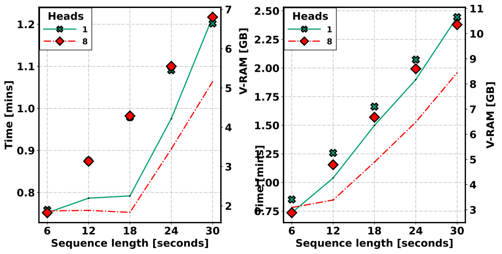


Figure 4.17: Forward pass of 1 and 8 heads for HyperConformer..

Figure 4.18: Overall time (minutes) and GPU consumption (GB) required by different architectures for sequences of different lengths. The left plot of (a) and (b) shows the small model, the right plot shows the mid-size model. Each sequence length in the x-axis represents 1000 samples from the LibriSpeech dataset. For all plots: Lines denotes time (left y-axis) and markers of GPU consumption (right y-axis). Batch size is 16 for all configurations.

4.4.5 Results & Discussion

Our experiments are designed to answer two questions: 1) Does HyperConformer perform competitive to Conformer in terms of word error rates? 2) Is HyperConformer more efficient than Conformer?

Speech recognition results We compare WERs of different state-of-the-art architectures for ASR, listing the results in Table 4.9. We find that HyperMixer alone achieves acceptable performance, especially in combination with a language model, but trails behind Transformer and Conformer, in all cases. We hypothesize that this is because the crucial local information in speech signals is difficult to pass through the hidden layer bottleneck of TM-MLP, which attention does not have. In contrast, HyperConformer performs comparable and often even better than Conformer in the medium-sized configuration. For instance, HyperConformer beats Conformer by 0.17% absolute WER on test-other with LM for the medium-sized model. We explain this as follows: In HyperConformer, i) the convolution module helps to model the local interactions between tokens, and ii) global interactions can be modeled in and passed through the multi-head HyperMixer's bottleneck effectively. Finally, we note that HyperConformer is amenable to scale, since moving from 7.9M → 21.7M, we obtain a 17.9% relative reduction in WER on test-other with LM, similar to Conformer.

Efficiency Analysis In [233] is shown that HyperMixer has efficiency benefits regarding processing speed and training data size. Here, we investigate if these properties also transfer to the speech domain, particularly, HyperConformer on the ASR task.

Peak memory consumption The right-hand side of Table 4.9 shows the peak memory consumption when training models of the same size on the same hardware. We observe that HyperConformer requires substantially less memory than Conformer (-30.6% with small size and -19.7% with medium size). The effect is stronger on small models than on large ones. Since larger models are wider (i.e., larger d and d'), the feature mixing components as well as $\mathrm{TM\text{-}MLP}$ require considerably more compute in comparison to attention, whose complexity depends primarily on the sequence length, which remains the same between training scenarios.

Resource consumption depending on sequence length The main advantage of HyperMixer is its linear complexity compared to attention's quadratic complexity. To investigate this property, we measure the peak memory and processing time of the encoder as a function of the length of the speech sample. To this end, we synthesize 1,000 sentences of 6, 12, 18, 24, and 30 seconds each by concatenating multiple signals from the LibriSpeech dataset. Figure 4.16 shows the resource consumption of all models. While HyperConformer and Conformer require similar processing time at short sequences, HyperConformer is considerably faster at mid-length (18s, small: 37.9%, mid: 15.2%) and long sequences (30s, small 56.1%, mid: 34.2%), demonstrating its better asymptotic complexity compared to Conformer. Note that Conformer with `regularMHA` is more efficient than `RelPosMHAXL`. However, this would lead to a performance loss [57], and HyperConformer is still substantially more efficient.

Number of heads An important technical novelty is the introduction of multi-head HyperMixer, which allows for multiple parallel views on the data analogous to multi-head attention, while at the same time reducing the model's complexity. In preliminary experiments, we found that HyperConformer with $k = 8$ heads performs as well as with $k = 1$ head. At the same time, moving from a single head to 8 heads reduces the number of parameters in the model by 7.1% in the small model and 20.8% in the mid-size model. Moreover, as Figure 4.17 shows, the processing time is reduced substantially by up to 12.6% (small) and 19.9% (mid-size) on the longest sequences.

Table 4.10: Performance of Conformer and HyperConformer when trained on 100h LibriSpeech (10× less data). Percentage in brackets shows relative WER reduction on test-other with LM.

Model	Small size	Medium size
Conformer	8.29	7.57
HyperConformer	6.76 (-18.5%)	5.80 (-23.4%)

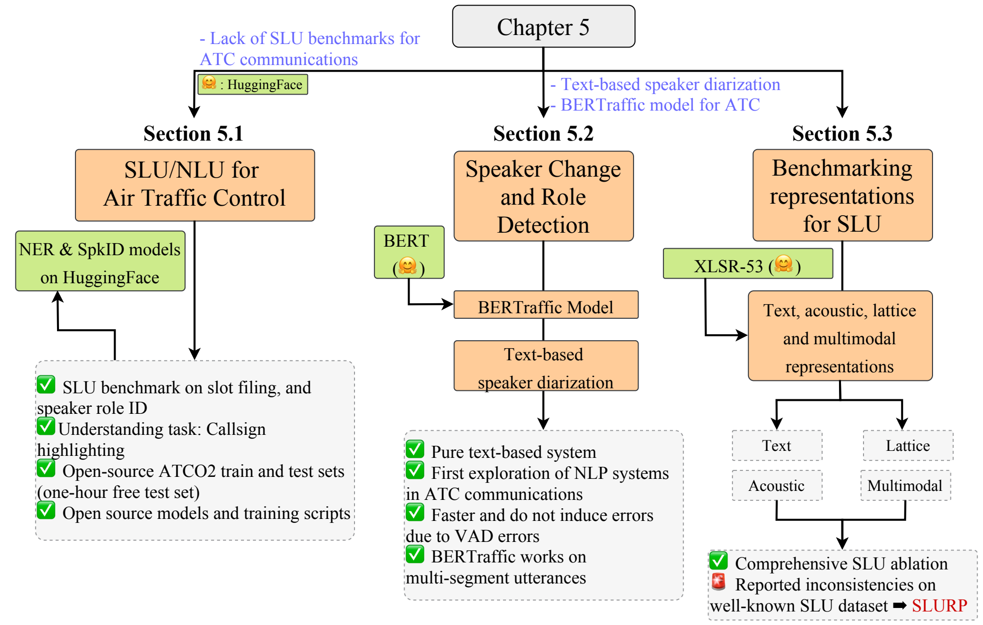
Low-resource scenario HyperMixer is reported to work better than MHSA in the low-resource scenario [233]. Here, we conduct an initial experiment to test whether HyperConformer inhibits the same characteristic. To this end, we compare HyperConformer to Conformer on the 100h LibriSpeech subset, which is 10 times smaller than the full dataset. All other training parameters remain the same. Table 4.10 shows the results. In this scenario, HyperConformer performs around 20% better than Conformer, suggesting better data efficiency.

Conclusions HyperConformer is a new architecture for efficient ASR introduced in this work. It integrates the benefits of the Convolution module from Conformer, which models local interactions, and the hypernetwork-based architecture, HyperMixer, which models global interactions. We were able to attain comparable or lower WERs (2.28/5.42 in test clean/other) HyperConformer when compared to Conformer. In addition, this novel architecture is substantially faster on long sequences, while also requiring less GPU memory during training. We believe HyperConformer is a green alternative to previous established Transformer and Conformer based models for ASR.

5 Towards Better Spoken Language Understanding

Introduction

This chapter focuses on natural and spoken language understanding (NLU/SLU). In complex scenarios and applications, the pipelines include SLU models that are fed with ASR outputs, thus (1) we tackle multiple NLU tasks, including slot filling and callsign highlighting for ATC (§ 5.1) and speaker change and speaker role detection (§ 5.2) for ATC. Furthermore, (2) we explore different modalities and representations that can be used for SLU, such as text, acoustic, lattice and multimodal (§ 5.3). An overall overview of this chapter is in Figure 5.1.


Chapter 5
- Lack of SLU benchmarks for ATC communications
🤗 : HuggingFace
- Text-based speaker diarization
- BERTraffic model for ATC
Section 5.1
SLU/NLU for Air Traffic Control
NER & SpkID models on HuggingFace
Section 5.2
Speaker Change and Role Detection
BERT (🤗)
BERTraffic Model
Text-based speaker diarization
Section 5.3
Benchmarking representations for SLU
XLSR-53 (🤗)
Text, acoustic, lattice and multimodal representations
Text
Lattice
Acoustic
Multimodal

✓ SLU benchmark on slot filing, and speaker role ID
✓ Understanding task: Callsign highlighting
✓ Open-source ATCO2 train and test sets (one-hour free test set)
✓ Open source models and training scripts

✓ Pure text-based system
✓ First exploration of NLP systems in ATC communications
✓ Faster and do not induce errors due to VAD errors
✓ BERTraffic works on multi-segment utterances

✓ Comprehensive SLU ablation
🚨 Reported inconsistencies on well-known SLU dataset ➡ SLURP

Figure 5.1: Overview of Chapter 5.

5.1 Spoken Language Understanding of Air Traffic Control Communications

Until the previous decade, research on air traffic control (ATC) communications was directed at only transcribing the dialogues between ATCos and pilots. However, transcription is only one intermediate task and further information, such as, entity highlighting (also known as intent and slot filling) or speaker role detection is imperative in real-life ATC control rooms. The process of parsing these high-level entities from ATC audio can be seen as SLU, or from text as NLU. In this work, we propose several approaches to handle ATC speech and extract high-level information and entities that can be used in downstream tasks.

Our contributions are covered below:

- We introduce the first open-source model in the field of ATC for slot filling based on a newly open sourced database;[a]
- we provide a comprehensive ablation of multiple SLU tasks for the field of ATC, including slot filling, ASR, speaker role detection and callsign highlighting;
- we introduce a 4h test set for the field of ATC that contains labels for ASR and multiple SLU tasks. A 1h open-source version (*ATCO2-test-set-1h*) that is available for free.[b]

Publication Note

The material presented in this section is adapted from the following publications:

- J. Zuluaga-Gomez, K. Veselý, I. Szöke, A. Blatt, P. Motlicek, M. Kocour, M. Rigault, K. Choukri, A. Prasad, S. S. Sarfjoo, *et al.*, "ATCO2 Corpus: A Large-Scale Dataset for Research on Automatic Speech Recognition and Natural Language Understanding of Air Traffic Control Communications," *Submitted to Data-centric Machine Learning Research (DMLR) Journal, arXiv preprint arXiv:2211.04054*, 2024
- J. Zuluaga-Gomez, A. Prasad, I. Nigmatulina, P. Motlicek, and M. Kleinert, "A Virtual Simulation-Pilot Agent for Training of Air Traffic Controllers," *Aerospace*, vol. 10, no. 5, p. 490, 2023
- J. Zuluaga-Gomez, K. Veselý, A. Blatt, P. Motlicek, D. Klakow, A. Tart, I. Szöke, A. Prasad, S. Sarfjoo, P. Kolčárek, *et al.*, "Automatic call sign detection: Matching air surveillance data with air traffic spoken communications," in *Proceedings*, vol. 59, no. 1. MDPI, 2020, p. 14

Supplementary materials related to this section:

- **Code - GitHub repository at:** https://github.com/idiap/bert-text-diarization-atc and in https://github.com/idiap/atco2-corpus.
- ATCO2 project website: https://www.atco2.org/

Major contributions Problem definition and experimental design and setup. Data preparation for ASR and NLU experiments. Trained and fine-tuned the NLP models for the experiments. Lead the work, including the papers write up.

[a] The full ATCO2 corpus is available for purchase through ELDA in http://catalog.elra.info/en-us/repository/browse/ELRA-S0484.

[b] This test set can be accessed for free in https://www.atco2.org/data.

5.1.1 Introduction

Previous work has explored different NLP tasks in the area of ATC. For instance, [111] describes a set of entities and elements that are present in ATC communications that are of special interest, e.g., commands and instructions [130]. The authors recommend that an operational system should be composed of an ASR module to obtain the word-level transcripts of the communication. Later, a subsequent system should extract ATC-related key entities and then parse them into a specific grammar. We redirect the reader to [196], which developed an ATC-structured grammar accepted by several European institutes. Furthermore, in [111], the process of extracting key entities from audio is summarized in an entire pipeline composed of three submodules. Namely, speaker role detection, intent classification and, slot filling (analogous to NER but on audio level). They aim at inferring the near-future air traffic dynamics, which can aid ATCos in their daily task. In addition, this system can notice communication errors caused by one of the speakers, also known as hear or read back errors. Some exploratory work addressing NLP and NLU on the framework of HAAWAII and ATCO2 projects (see Table 2.2) is described in [34, 14].

In this section, we describe our baselines for two tasks related to NLP and NLU.[1] In air traffic control applications, in addition to transcripts generated by an ASR system, we can also extract rich metadata from the transcripts and audio. Some examples are–but not limited to–:

- ✓ What are the high-level entities in the communication? → named-entity recognition (NER) or slot filling (SF). Previous work is presented in [15].
- ✓ Who is talking? ATCo or pilot → speaker role detection (SRD), sequence classification. Early work is presented in [34].
- ✗ Is the pilot responding the correct information? → read-back error detection. Previous work is presented in [24] under HAAWAII project, and in [31] funded by SESAR 2020 PROSA project (PJ.10-W2), as well as in some others submissions [254, 25],
- ✗ Is the communication being uttered in English language? → English language detection (ELD). Previous work is presented in [125].

We present baselines only on the above items marked with ✓, while the items marked with ✗ are, either covered in previous work or left as future research directions. Generally speaking, extracting the above-mentioned information could allow to further fulfill other ATC tasks, e.g., pre-filling radar labels in the ATC control rooms. Or, for example, reduce the workload of ATCos and make them more efficient by automating manual and hard work. Also, this leads to reduced overall probability of incidents and accidents due to air traffic management erroneous procedures.

5.1.2 Slot Filling & Named Entity Recognition

Named entity recognition, or NER, is one of the most explored tasks in the field of information extraction and NLP [255]. NER aims to locate and classify entities in unstructured text into

[1] As we work on top of ASR transcripts, these tasks can be also cataloged as SLU.

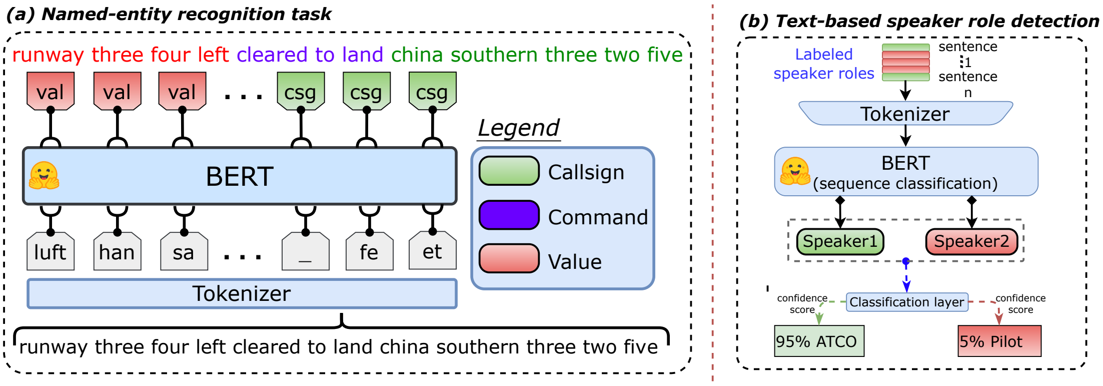

(a) Named-entity recognition task
runway three four left cleared to land china southern three two five
val
val
val
. . .
csg
csg
csg
BERT
luft
han
sa
. . .
_
fe
et
Tokenizer
runway three four left cleared to land china southern three two five
Legend
Callsign
Command
Value
(b) Text-based speaker role detection
Labeled speaker roles
sentence
1
sentence
n
Tokenizer
BERT
(sequence classification)
Speaker1
Speaker2
confidence score
Classification layer
confidence score
95% ATCO
5% Pilot

Figure 5.2: (a) Named entity recognition (or Slot filling) and (b) speaker role detection based on sequence classification (SC) for ATC utterances. Both systems fine-tune a pretrained BERT [87] model for ATC tasks. The NER systems recognizes callsign, command and values, while the SC assigns a speaker role to the input sequence.

pre-defined classes or categories. Examples are, persons or organization names, expressions, or, for instance, callsigns or commands in ATC (see Figure 5.2). Initially, NER was based on handcrafted lexicons, ontology, dictionaries, and rules [256]. Even though these systems were interpretable and understandable, they were prone to human errors. Collobert et al. [93] introduced machine learning-based methods for text processing in topics such as part-of-speech tagging, chunking, NER, and semantic role labeling. Further interesting works on NER are [91] focusing on multilingual NER for Slavic languages, and [92] presenting a broad survey of NER methods. In practice, a NER system can be crafted by fine-tuning a pretrained LM, e.g., BERT [87], RoBERTa [88], or DeBERTa [89]. Nonetheless, these models are data hungry and need expensive GPUs during its training and inference. Further work has been directed at reducing their computational footprint, by performing, for example, knowledge distillation [257].

Air traffic control communications frequently carry structured information. A typical ATCo-pilot utterance consists of three major entities: The Callsign as plane-identifier, which is followed by the command and a value that specifies the command further. An example of the entities is shown in Figure 5.2. Furthermore, an example of a tagged transcript is as follows:

```
<COM> CLIMBING TO </COM> <VAL> FLIGHT LEVEL SEVEN ZERO </VAL>
<CAL> OSCAR KILO TANGO UNIFORM ROMEO </CAL>
```

For the labeling the *IOB* format is used, which stands for *inside*, *outside* and *beginning*. This results in the labels B-CALL, I-CALL and O (same for COM and VAL). The label B-CALL marks the beginning of the callsign in the transcript, while I-CALL labels are used for words of the callsign that are inside the named entity. All words that are outside a callsign are marked with the O tag or the other classes. The task of the NER module is to produce the correct label for each word in the transcript. The correct labeling of the transcript above will look as follows:

"**B-com I-com B-val I-val I-val I-val B-call I-call I-call I-call I-call**"

The *ATCO2-test-set corpus* provides transcription on the word level that assigns pieces of text to these specific classes. We developed a baseline system to extract such information from ASR outputs, as depicted in Figure 5.2. An early implementation of this system was covered in [15]. However, these experiments were carried over private databases, so it is difficult to compare with our current results. This is the main reason of open sourcing scripts to fine-tune a NER model with the version of *ATCO2-test-set corpus*.[2]

Experimental Setup

Our experiments are carried out with *ATCO2-test-set corpus* only, for both, training and evaluation.[3] The main reason is that none of the public databases from Table 2.2 contain NER transcriptions. As a workaround, we implemented a simple k-fold cross-validation scheme. We define $K = 5$ folds, with a 70/10/20 ratio for train/dev/test subsets, respectively. We use ground truth transcripts for training and testing NER.

We download BERT[4] [87] from HuggingFace [155, 156]. We append a linear layer with a dimension of 7 on top of the last layer of the BERT model.[5] The model is later fine-tuned on the NER task, with each Fold K of the train splits. Each model is fine-tuned on an NVIDIA GeForce RTX 3090 for 10k steps. During experimentation, we use the same learning rate of $\gamma = 5\text{e}{-5}$ with a linear learning rate scheduler. Dropout [157] is set to $dp = 0.1$ for the attention and hidden layers, while Gaussian Error Linear Units (GELU) is used as activation function [158]. We also employ gradient norm clipping [258]. We fine-tune each model with an effective batch size of 32 over 50 epochs with AdamW [159] optimizer (β_1=0.9, β_2=0.999, ϵ=$1\text{e}{-8}$).

Results

We report the results obtained from the 5-fold cross validation experiments. We split the results by tags, i.e., callsign, command and values. For each of them, we report precision, recall and F1-scores in Table 5.1. We obtained an average of 0.97, 0.82 and 0.87 F1-score for callsign, commands and values. We observed that the command class was the most challenging among the three classes. We believe this is because commands contain extra complexity in comparison to callsigns and values. For example, in some cases the ATCos or pilots use several commands, or these are sometimes mixed in the same utterance. In contrary, callsigns follow a standard form, composed of an airline designator, numbers, and letters (spelled in ICAO phraseology). Values are composed of cardinal numbers and some standard words, e.g., 'flight level'. We also noted a

[2] See the model in https://huggingface.co/Jzuluaga/bert-base-ner-atc-en-atco2-1h.

[3] We provide in the GitHub repository the utterance IDs splits utilized for these experiments.

[4] We use the pretrained version of `bert-base-uncased` with 110 million parameters for all the experiments.

[5] Following the Inside–outside–beginning (IOB) format, i.e., two outputs for each NER class. In this case, we have callsign, command and values. an extra class for the "outside" tag, i.e., none.

Table 5.1: Different performance metrics for callsign, command and values classes of the NER system. Metrics reported for each of the 5-fold cross-validation scheme on *ATCO2-test-set corpus* with a `bert-base-uncased` model. @P, @R, and @F1 refer to precision, recall and F1-score, respectively. Numbers in **bold** refer to the top performance per column among folds. †mean score over the 5 folds.

Fold	**Callsign**			**Command**			**Values**		
	@P	@R	@F1	@P	@R	@F1	@P	@R	@F1
1	0.97	0.98	0.97	0.80	0.81	0.81	0.86	0.86	0.86
2	0.97	0.98	0.97	**0.83**	**0.86**	**0.85**	0.86	0.89	0.87
3	0.97	0.97	0.97	0.81	0.85	0.83	**0.87**	0.87	0.87
4	**0.98**	**0.98**	0.98	0.78	0.80	0.79	0.85	**0.90**	0.87
5	0.97	0.98	**0.98**	0.80	0.83	0.81	**0.87**	0.89	**0.88**
AVG†	0.97	0.98	0.97	0.80	0.83	0.82	0.86	0.88	0.87

significant irregularity in performance for the command class between the 5 folds (see column: Command in Table 5.1). For example, worse → best scenario on F1-score was 0.79 → 0.85, almost a six-point drop. A five-point drop is also seen in precision and recall. These results are seen when comparing fold 2 (best) against fold 4 (worst).

In conclusion, the results from Table 5.1 are the first official baseline for NER[6] on the *ATCO2-test-set corpus*. However, there is room for improvement. For instance, implementing semi-supervised learning or data augmentation should bring robustness and yield higher performance. Similarly, one can pretrain the LM directly on ATC text rather than standard English text, which should bring in additional benefits. We leave this line of research for future work.

5.1.3 Callsign Recognition and Understanding

The named entity recognition system is capable to select words which form a callsign (i.e., highlight 'swiss two six eight nine'). However, *ICAO Callsign Extraction* produces the callsign directly in ICAO format (e.g., SWR2689), which is more useful for applications. This is not trivial because callsigns get commonly shortened, if the situation is obvious (e.g., 'swiss two six eight nine' → 'six eight nine', or 'swiss eight nine'). And the underlying ASR produces errors in its automatic transcripts. In this work, we explored two approaches. In [259], the ICAO callsign is retrieved by a BERT-based Encoder-Decoder neural network. This system directly takes outputs from an in-domain ASR system and extracts the ICAO callsign without relying on Named Entity Recognition as an intermediate step. The model uses a list of callsigns, i.e., context information, to predict the callsign in ICAO format. The advantage of this sequence-to-sequence approach is, that it does not just select the best callsign from the surveillance list, but it can also

[6] After extensive research, to authors' knowledge, this is the first official baseline on NER for air traffic control communications. We have not found any other work that is both, open-source and that targets NER.

extract unknown callsigns, that are not present in the initial list. The overall approach is depicted in Figure 5.3.

The second approach–covered in [15]–performs NER to extract the callsign within the sentence, which is later ranked by Levenshtein distance with the ones in the callsign list from the surveillance data. This approach always selects a callsign from the list. We showed that boosting callsigns with the combination of ASR and NLP methods eventually leads up to 53.7% of an absolute, or 60.4% of a relative, improvement in callsign recognition.

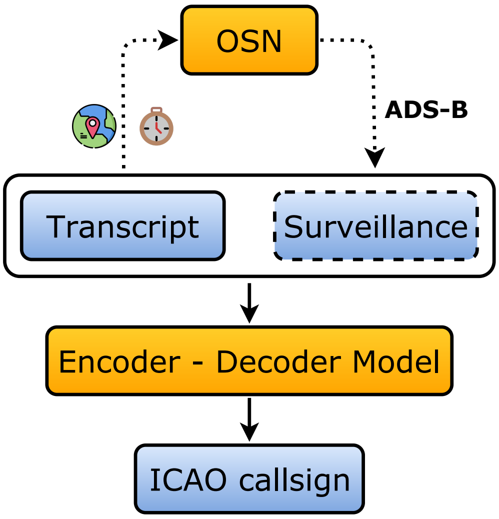


Figure 5.3: Proposed callsign recognition and understanding system. The dotted path marks the optional surveillance retrieval via OpenSky Network (OSN) with the aid of the transcripts timestamp and VHF receiver location. Taken from [259].

5.1.4 Speaker Role Detection

In NLP, text classification or sequence classification (SC) is a task that assigns a label or a class to a sequence of words [260, 261]. The hypothesis is that the words within the given text share a common role and meaning inside the sentence's grammatical structure. One of the most acknowledged forms of SC is sentiment analysis, which assigns a label like positive, negative, or neutral to a sequence of text embeddings [262]. Nowadays, state-of-the-art SC systems are based on the well-known Transformer, e.g., BERT [87] or RoBERTa [88]. Akin to NER, SC is considered a downstream task operating on ASR output.

In ATC, the dialogues are built on top of a well-defined lexicon and dictionary, which follows a simple grammar. This standard phraseology has been defined by the ICAO [263] to guarantee the safety and reduce miscommunications between the ATCos and pilots. In this work, we propose some baselines on the SC task aimed at detecting the speaker role from transcribed ATC communications (sentences). Our previous work on speaker role detection is covered in [14, 34].

Experimental Setup

The SC experiments are similar to the ones in NER (see above). Specifically, we use the same model (`bert-base-uncased`), hyperparameters (e.g., number of epochs), optimizer, dropout rates, etc. However, here, we fine-tuned the model on the SC task rather than NER. We append a linear layer with a dimension of 4 on top of the last layer of the BERT model, i.e., a two-class

Table 5.2: Different performance metrics for the speaker role detection experiments. Metrics reported on *ATCO2-test-set corpus* with a `bert-base-uncased` model. @P, @R, and @F1 refer to precision, recall and F1-score, respectively. Numbers in **bold** refer to the top performance per column.

Training Corpus	ATCO			PILOT			AVG
	@P	@R	@F1	@P	@R	@F1	@F1
LDC-ATCC	0.87	0.73	0.79	0.70	0.86	0.77	0.78
UWB-ATCC	0.88	**0.83**	**0.86**	**0.80**	0.85	0.82	**0.84**
LDC-ATCC + UWB-ATCC	**0.92**	0.78	0.85	0.76	**0.91**	**0.83**	**0.84**

classification model.[7] Here, each word is assigned a tag as belonging to ATCo or pilot, thus, we have "B-atco", "I-atco", "B-pilot", and "I-pilot" tags.

We employed LDC-ATCC[8] and UWB-ATCC[9] datasets for fine-tuning and *ATCO2-test-set corpus* for testing. In LDC-ATCC and UWB-ATCC databases, speaker roles tags for each sample are marked in the original transcripts. And, we use ground truth ASR transcripts the evaluation. We create speaker-independent train/test splits based on the original databases. The split IDs for each subset are registered in the public GitHub repository of this paper.

Results

We report the baseline results for speaker role detection in Table 5.2. Differently from NER, we only used *ATCO2-test-set corpus* for evaluation. We trained three models using different training datasets. From Table 5.2 we can see that pilots' communications are more challenging for our model in comparison to the ones from ATCos. For instance, in the model fine-tuned with LDC-ATCC corpus, there is a two-point drop in F1-scores for pilots, i.e., 0.79 → 0.77 F1-score. Similar behavior is seen in the model fine-tuned with UWB-ATCC corpus, i.e., a four-point drop in F1-scores, 0.86 → 0.82. However, models trained on the later show more robustness for both classes in comparison to the one trained with LDC-ATCC.

We also investigated the performance benefit of combining both datasets. For this experiment, we only obtained one point increase for the pilot class, while one point decrease for the ATCo class, both in comparison to the model trained on UWB-ATCC only. It is important to keep in mind that *ATCO2-test-set corpus* is a completely unseen dataset throughout all the experiments. We

[7] Seee further details in the open-source repository: https://github.com/idiap/atco2-corpus.

[8] The Air Traffic Control Corpus (LDC-ATCC) corpus is public in: https://catalog.ldc.upenn.edu/LDC94S14A. It comprises recorded speech for use in the area of ASR for ATC. The audio data is composed of voice communication traffic between various controllers and pilots.

[9] The UWB-ATCC corpus is released by the University of West Bohemia, and it can be downloaded for free in: https://lindat.mff.cuni.cz/repository/xmlui/handle/11858/00-097C-0000-0001-CCA1-0. The corpus contains recordings of communication between ATCos and pilots. The speech is manually transcribed and labeled with the speaker information, i.e., whether ATCo or pilot is speaking and when.

are convinced that integrating a small in-domain development set could boost the performances. Further research towards text-based speaker diarization and speaker role detection is carried in [4]. Here, we study the SC task for ATC with different models architectures, including BERT [87], RoBERTa [88] and DeBERTa [89].

Conclusions This work represents a significant step forward in the field of ATC. Here, we explore NLU/SLU based approaches to extract the high-level information from ATCo-pilot dialogues. We have introduced innovative models and extensive resources, including the first open-source model for slot filling in ATC communications and a detailed ablation study across multiple SLU tasks. Our contributions show that it is possible to enhance the operational efficiency of ATC systems while reducing the risk of miscommunication and increasing overall airspace safety. This integrated approach of ATC understanding could significantly alleviate the workload of ATCos. Finally, we open sourced several models, databases, and scripts to replicate our results, which is already making an impact in the community.

5.2 Text-Based Joint Speaker Role & Speaker Change Detection

Automatic speech recognition (ASR) allows transcribing the communications between air traffic controllers (ATCos) and aircraft pilots. The transcriptions are used later to extract ATC named entities, e.g., aircraft callsigns. One common challenge is speech activity detection (SAD) and speaker diarization (SD). In the failure condition, two or more segments remain in the same recording, jeopardizing the overall performance. We propose a system that combines SAD and a BERT model to perform speaker change detection and speaker role detection (SRD) by chunking ASR transcripts, i.e., SD with a defined number of speakers together with SRD. The proposed model is evaluated on real-life public ATC databases.

Publication Note

The material presented in this section is adapted from the following publication:

- J. Zuluaga-Gomez, S. S. Sarfjoo, A. Prasad, I. Nigmatulina, P. Motlicek, K. Ondrej, O. Ohneiser, and H. Helmke, “BERTRAFFIC: Bert-based joint speaker role and speaker change detection for air traffic control communications,” in *IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 633–640

Supplementary materials related to this section:

- **Code - GitHub repository at:** https://github.com/idiap/bert-text-diarization-atc
- **Models at HuggingFace:** https://huggingface.co/Jzuluaga, ID for speaker role: `bert-base-speaker-role-atc-en-uwb-atcc` and `bert-base-token-classification-for-atc-en-uwb-atcc`; ID for NER: `bert-base-ner-atc-en-atco2-1h`

Major contributions Problem definition and experimental design and setup. Data preparation for experiments on text-based speaker diarization. Trained and fine-tuned the NLP models for the experiments. Lead the work, including the paper write-up.

5.2.1 Introduction

Automatic speech recognition (ASR) allows transcribing the communications between air traffic controllers (ATCos) and aircraft pilots. The transcriptions are used later to extract ATC named entities, e.g., aircraft callsigns. One common challenge is speech activity detection (SAD) and speaker diarization (SD). In the failure condition, two or more segments remain in the same recording, jeopardizing the overall performance. We propose a system that combines SAD and a BERT model [87] to perform speaker change detection and speaker role detection (SRD) by chunking ASR transcripts, i.e., SD with a defined number of speakers together with SRD. The proposed model is evaluated on real-life public ATC databases. Previous work concluded that higher accent variability and noise level cause ASR systems to yield up to two times higher word error rates (WER) for pilots’ utterances compared to ATCos’ utterances [142]. In addition, close and cross-talk between ATCo and pilots induce traditional speaker diarization (SD) systems to yield low performances. Thus, this jeopardizes the speaker change detection (SCD) step and subsequently the ASR system ends up processing utterances with multiple speakers.

Motivation Already existent acoustic-based SD systems, like [264] or end-to-end neural-based SD [265], show promising performances for many applications. However, in ATC communications, given its limitations such as high speaker rate, close-talk, and noise levels, relying solely on the acoustic level has shown to be insufficient. Additionally, standard SD systems add one layer of complexity to the whole ATC pipeline,[10] weakening the flexibility to transfer the already tuned pipelines to other environments. Thus, text-based SD stands as an interesting solution.

Table 5.3: Conversation between two speakers with correct SAD and SCD (rows 1 and 2) and SCD fault (row 3, words in bold). †samples from *SOL-Cnt* test set.

Speaker Label	Detected segment†
ATCo (spk. 1)	<s> november six two nine charlie tango report when established </s>
Pilot (spk. 2)	<s> report when established november six two nine charlie tango </s>
Mixed (SAD and SCD failed)	<s> november six two nine charlie tango report when established **report when established** </s> <s> november six two nine charlie tango </s>

Contribution In this work, we fine-tune a pre-trained BERT model [87] to jointly perform tagging and chunking. Chunking allows splitting sentences into tokens (or words) and then merging them in meaningful subgroups. Here, a phrase (or entity) is composed of a full single-speaker utterance of ATCo or pilot (see Table 5.3). By applying chunking in a multi-speaker and multi-segment utterance, one can perform speaker change detection (SCD) and speaker role detection (SRD) simultaneously on the text level (Figure 5.4 mid-box). The proposed approach simplifies the standard SD pipeline, moving up the task from the acoustic level to text level, i.e., post ASR. We stack the BERT model on top of a speech activity detection (SAD) module to create a text-based SD, which from now on we call ***'BERT SD system'***. Speaker diarization systems answer the question *"who spoke when?"*. SAD, segmentation or SCD, embedding extraction, clustering and labeling are the main parts of a SD system. An overview of acoustic-based diarization is covered in our previous work [14], including state-of-the-art approaches based on DNNs, i.e., end-to-end neural diarization (EEND) [265, 266, 267, 268].

Text-based speaker role detection Early text-based techniques for SRD or SCD relied on handcrafted lexicons, dictionaries, and rules. They are prone to human errors and not robust against noisy labels, e.g., produced by standard ASR systems. Collobert et al. [93] introduced machine learning methods for text processing in part-of-speech tagging, chunking, and semantic role labeling. In this work, we employ chunking, which means tagging and splitting an ATC utterance. This allows us to perform jointly SCD and SRD from text. In [269] a text-based SRD for multiparty dialogues is proposed, but limited to SRD. Text-based diarization has been proposed in the past by [270, 271]. However, these previous works do not take into account the text structure, grammar, and syntax.

[10] A standard ATC pipeline is composed of signal processing, SAD and SD, ASR and NLU.

Contrasting with previous work Different to other systems, e.g., EEND or traditional acoustic-based SD, our model is fed with text inputs only (e.g., ASR transcripts). The field of ATC shows limitations and advantages w.r.t acoustic-based EEND. Limitations: ATC audio is noisy (below 15 dB SNR) with close and cross-talk speech. Advantages: the number of speaker roles are known and the ATC grammar is well defined, albeit it differs between ATCo and pilots (e.g., speaker roles). Thus, we leverage those advantages in order to show that a fully text-based joint SCD and SRD system can perform on par or even better than traditional acoustic-based SD.

5.2.2 Databases and Experimental Setup

Databases and annotation protocol We use a mix of private and public database as describe in Table 5.4.[11] In addition to manual speech transcripts, speaker labels and time segmentation (e.g., ATCo/pilot/mixed) are also available. The BERT model starts by tagging each word of the transcript (ground truth or ASR transcript) with a set of tags that follows the well-known *IOB format* (Inside-Outside-Beginning). In IOB format, each entity (a full sentence in our case) is composed of two tags: (i) the *Beginning* defines which token/word is the start of the sentence ***'B-'***, and (ii) the *Inside* tag ***'I-'*** defines which tokens/words belongs to that specific sentence. We define ATCo recordings as *Speaker1*, while pilot segments as *Speaker2* (green and red in Figure 5.4, respectively). We do not use the *Outside* tag, i.e., we know that each word belongs to one class.

Table 5.4: Train and test statistics per database. *ATCo* and *pilot* columns: single-speaker segments. *Mixed* column: samples with two or more segments. †real-life ATC set with SAD failure.

Database	ATCo	Pilot	Mixed	Ref
Private databases				
SOL-Cnt†	662 / 138	945 / 204	535 / 205	[272]
HAAWAII	18724 / 1954	21099 / 2299	- / -	[13]
Public databases				
ATCO2	- / 1772	- / 1350	- / -	[22]
LDC-ATCC	12694 / 1515	14216 / 1446	- / -	[121]
UWB-ATCC†	4577 / 1157	6669 / 1713	735/174	[120]

Data augmentation We implemented a simple yet effective data augmentation pipeline to counteract the class imbalance in the train sets (see Table 5.4). First, we split the training sets on either *ATCo* (speaker 1) or pilot (speaker 2) subset. Then, we generate new sentences from the initial set of utterances for each database (e.g., HAAWAII ∼39k utterances). Each new sample depends on: (i) the number of sentences to be concatenated, and (ii) the speaker label of each sentence. New samples are composed of one to four sentences, each with an equal chance of being drawn from the ATCo or pilot dictionary. We generate ∼350 MB of text data

[11] **SOL-Cnt:** a private database recorded and collected over EU-funded industrial research project that aims to reduce ATCos' workload with an ASR-supported aircraft radar label. Voice utterances of ATCos and pilots have been recorded in the operations' room at the ANSP site of Austrocontrol in Vienna, Austria. PJ.16-W1-04 project [272]: https://www.sesarju.eu/projects/cwphmi.

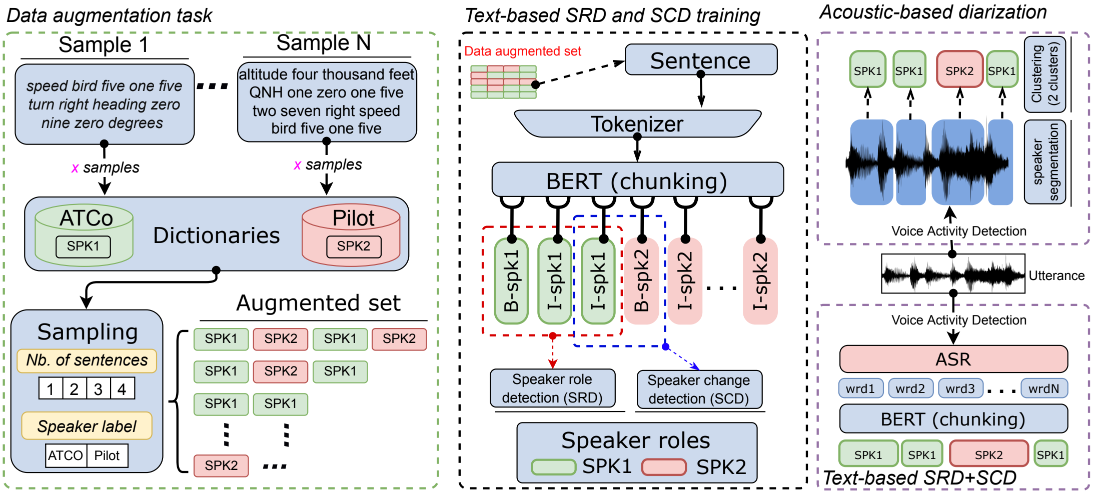


Figure 5.4: **Left block:** data augmentation pipeline. Augmented samples contain between one and four utterances (probabilities of 40%, 30%, 20% and 10%). New sentences have equal chance to be sampled from the ATCo or pilot dictionary. **Central block:** BERTraffic pipeline. **Right block:** pipeline to compare acoustic (VBx) and text-based joint SRD and SCD.

(∼1M sentences) in this way. We emphasize that in ATC, there is no need to have a correlation between previous and next sentences/utterances. This is due to the fact that speaker 1 (ATCo) communicates to several speakers 2 (pilots). The stream of information received and transmitted by the speakers is not dependent on 'left' or 'right' context. Therefore, concatenating various segments randomly would not degrade substantially the WERs.[12] The left block in Figure 5.4 depicts the proposed data augmentation pipeline.

5.2.3 BERTraffic System

The performance of our BERT-based SRD and SCD system is contrasted with a standard acoustic-based SD system. We use an out-of-the-box VBx system to evaluate the *SOL-Cnt* and *UWB-ATCC* test sets, which contain real-life ATC audio where segmentation failed. For both, BERT and acoustic-based SD systems, we use the same multilingual ASR-based SAD module [273] to remove the silence in the recording files.

Speaker role and speaker change detection module The SRD and SCD systems are built on top of a pretrained BERT model [87] downloaded from HuggingFace [155, 156]. The model is later fine-tuned with either the original or the augmented databases, on the tagging and chunking task (following *IOB* format). We append a linear layer with a dimension of 4 on top of the last layer of the BERT model, i.e., we use the same classes and tags in Section 5.1. Then, we fine-tune each model on a RTX 3090 for 3k steps, with a learning rate scheduler that first warms up the

[12] We measure WERs on the original and augmented test sets. The relative WER degradation is less than 1%.

learning rate until $\gamma = 5\text{e}{-5}$ for 500 steps, and then it linearly decays. We employ AdamW [159] optimizer (β_1=0.9, β_2=0.999, ϵ=1e−8) and dropout [157] of $dp = 0.1$ for the attention and hidden layers. We use GELU activation function [158]. We train all models with an effective batch size of 64.

Acoustic-based diarization module For details of the VBx model, the reader is referred to [264]. This model uses a Bayesian hidden Markov model (BHMM) to find speaker clusters in a sequence of x-vectors. Here, the x-vector extractor uses DNN architecture based on ResNet101. The input to the ResNet is 64 log Mel filter bank features extracted every 10 msec using 25 msec window. In the first step, Agglomerative Hierarchical Clustering (AHC) is applied to the extracted x-vectors. Then, Variational Bayes HMM over x-vectors is applied using the AHC output. For achieving the best performance on the database with short duration files with a maximum of two speakers, we tuned the probability of not switching speakers between frames (loopP) and speaker regularization coefficient (Fb) to 0.7 and 6, respectively.

Automatic speech recognition We train a hybrid-based ASR system tailored for ATC speech with Kaldi toolkit [100]. The system follows the standard recipe, e.g., uses MFCC and i-vectors features with standard chain training based on lattice-free MMI. Similar as in Section 3.1. We use the same ASR system for both speakers roles, ATCo and pilot. The training recipe and databases (including the training sets are Table 5.4) are covered in [22, 19, 29, 15].

5.2.4 Evaluation protocol

The experiments are prepared to answer three questions: (i) how reliable is the BERTraffic on SRD and SCD system on ground truth transcripts? (ii) How is the performance impacted when using automatically generated (ASR) transcripts instead of ground truth transcripts?[13] And, (iii) which system performs better on real-life ATC speech data, text or acoustic-based SD?

Evaluation: First, for ***acoustic-based diarization*** we use DER and Jaccard Error Rate (JER) as metrics. See more information in the SD subsection of Section 2.4. Second, for ***Speaker role detection***, we use with JER on the token level (which is more aligned to SD) on the five proposed test sets. **SOL-Cnt** and **UWB-ATCC** databases have utterances with more than one speaker per utterance. Thus, we test BERTraffic on SD on these two test sets. We first analyze the model performance on the ideal case, i.e., we used the ground truth audio annotations to obtain JERs per test set, thus assuming we have access to a perfect ASR system (0% WER). We employ the Scikit-learn[14] Python library to calculate these scores. Third, ***for speaker change***

[13] This is a real-life scenario where ASR transcripts are fed to the BERT SD system instead of ground truth transcripts.

[14] We use weighted Jaccard error rate score. It calculates metrics for each class (i.e., ATCo and pilot), and finds their average weighted by support (the number of true instances for each class). This accounts for label imbalance.

Table 5.5: Token-level JER from predictions using different train (column 1) and test sets. We report the mean ± STD across five training runs with different seeds. **Bold:** best performance over public databases. Underline: the highest performance per column.

Model		Test: public databases			Test: private database	
Database	**# samples**	**ATCO2**	**UWB-ATCC**	**LDC-ATCC**	**HAAWAII**	**SOL-Cnt**
Evaluation: public databases						
LDC-ATCC	26.9k	31.3 ± 2.4	35.8 ± 2.0	8.1 ± 0.7	28.7 ± 3.1	52.6 ± 1.3
UWB-ATCC	11.2k	21.6 ± 0.7	**10.7 ± 0.6**	18.7 ± 2.6	15.2 ± 1.4	**18.7 ± 1.7**
↪ + LDC-ATCC	38.1k	**19.8 ± 0.9**	11.3 ± 0.4	**7.1 ± 1.3**	**14.2 ± 1.4**	24.0 ± 1.9
Evaluation: private databases						
HAAWAII	39.8k	23.9 ± 0.6	22.3 ± 1.7	14.1 ± 1.2	6.5 ± 0.7	48.5 ± 1.4
↪ +LDC+UWB	77.9k	17.5 ± 0.2	11.5 ± 0.5	7.5 ± 0.6	6.2 ± 0.3	26.8 ± 2.0

detection we use DER and JER on one private (**SOL-Cnt**) and one public (**UWB-ATCC**) test set, which contains utterances with one or two speakers. For creating the segments from the BERTraffic SCD system, we used forced alignment between audio and ground truth text using the ASR module. Also, time information from the ASR output transcripts are used to create the segments of the BERTraffic SD system on the ASR transcripts.

5.2.5 Results & Discussion

Baseline performance of BERT SD We discuss the results listed in Table 5.5. Here, we aim at evaluating two aspects of the BERT SD system. First, we assess how well the model behaves on out-of-domain corpora. We fine-tune BERT models on each database and evaluate it on all five test sets. We call this: *transferability* between corpora. Second, we establish baselines on both, public and private databases. Each model is fine-tuned five times with different seeds, hence we report the mean and standard deviation across runs. Not to our surprise, test data that matched the fine-tuning one performed particularly well. LDC-ATCC and UWB-ATCC test sets reached less than 10% JER, while ∼20% JER for ATCO2.

One aspect that can shed light on new research is how public databases transfer to private ones. This can help future research to set a starting point, thus reducing the costs inherit by developing tools from scratch, e.g., SD system for ATC. We noted that UWB-ATCC corpus was more challenging for the BERT SD model compared to LDC-ATCC and HAAWAII corpora (6.5% and 8.1% JER, respectively). However, this system performed consistently better on all the other test sets, if we compare the model fine-tuned on UWB-ATCC versus the ones on LDC-ATCC and HAAWAII. We believe that the transferability to new domain of UWB-ATCC corpus is higher compared to LDC-ATCC and HAAWAII (see ‘UWB-ATCC’ row in Table 5.5 and compare it with LDC-ATCC or HAAWAII).

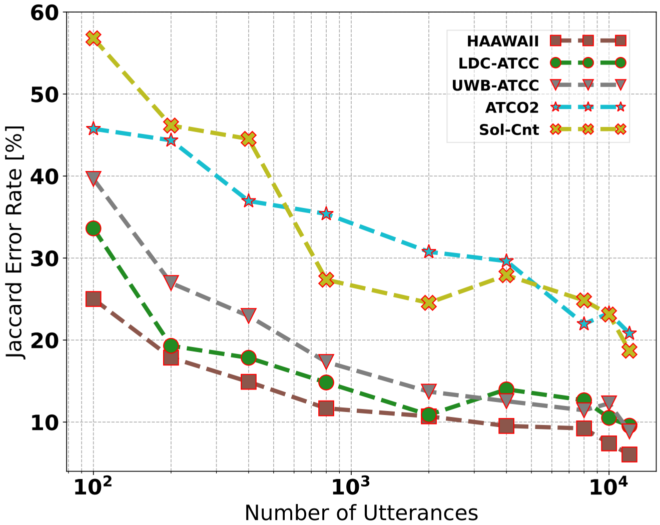


Figure 5.5: JER for nine models fine-tuned with increased amount of samples per database. We evaluate models on two configuration. (1) *in domain experiments* (HAAWAII, LDC-ATCC and UWB-ATCC) and (2) *out of domain test sets* (ATCO2 and SOL-Cnt). For the two later (blue and yellow dashed lines), we report the results of the model fine-tuned with UWB-ATCC database.

Does adding more data help? Here, we evaluate the BERT SD system by performing an ablation where the amount of fine-tuning data is incremental. In total, 9 models per database are studied, as depicted in Figure 5.5 (each data point represents one model). We report token-based JERs which are more aligned to standard SD. For the public databases, we obtained 65, 43, and 37% relative improvement in JERs on LDC, UWB, and ATCO2, respectively, by scaling up the fine-tuning data from 100 to 2000 samples. This number goes up to more than 50% relative JERs improvement if we use 10k samples (69% relative improvement for LDC). We note the same behavior on the private databases. At least 50% relative improvement is seen by scaling up the data from 100 samples → 2000 samples, on both, HAAWAII and SOL-Cnt experiments. To our surprise, UWB-ATCC models transfer particularly well on the two out-of-domain test sets (i.e., ATCO2 and SOL-Cnt). This gives insights that our approach works well on both, public and private databases. We believe this is an acceptable starting point for the future research on text-based SRD and SCD (not only aligned to ATC).

Robustness of BERT speaker diarization on ASR transcripts We evalute the BERT SD system on *SOL-Cnt* and *UWB-ATCC* test sets, which contain utterances with more than one speaker (*mixed* subset). We feed BERTraffic with 1-best transcripts obtained from our in-domain hybrid-based ASR system [15]. Table 5.6 lists the results with an additional line for *'ASR output'*. In the single-speaker case (either ATCo or pilot), the degradation (ASR transcripts instead of ground truth text) in SD from the BERT SD was no more than 1% absolute JER and DER (worse, Pilot subset 2.4 → 3.7% DER reduction in *SOL-Cnt* set). In the **MIXED** case, the degradation varied between 0.1% JER and 0.6% DER absolute in *SOL-Cnt* set, and 3.5% JER and 1.1% DER

Table 5.6: Comparison of acoustic VBx and text-based SD on *ATCo*, *PILOT*, and *MIXED* subsets of SOL-Cnt and UWB-ATCC test sets. **Bold:** top performance. †acoustic-based SD. ††BERTraffic trained on all corpora with data augmentation and evaluated on ground truth (_GT) or ASR outputs (_ASR).

	Sol-Cnt test set		**UWB-ATCC test set**	
	DER (%) $\downarrow$	**JER (%)** $\downarrow$	**DER (%)** $\downarrow$	**JER (%)** $\downarrow$
Model	AT / PI / MIX	AT / PI / MIX	AT / PI / MIX	AT / PI / MIX
Acoustic-based speaker diarization				
Acoustic_aIB†	14.8 / 13.9 / 13.1	15.6 / 13.5 / 25.5		
Acoustic_VBx†	5.8 / 7.8 / 10.3	7.0 / 10.9 / 22.2	**0.8** / **1.2** / 14.4	**0.6** / **0.7** / 39.4
Text-based speaker diarization				
BERT_GT††	**2.4** / **2.4** / **8.9**	**1.0** / **2.2** / **15.0**	1.2 / 1.7 / **5.8**	1.1 / 1.1 / **16.6**
BERT_ASR††	3.0 / 3.7 / 9.5	1.5 / 3.2 / 15.1	1.6 / 1.5 / 6.9	1.2 / 1.2 / 20.1

absolute in *UWB-ATCC* set. This behavior is mainly due to the noisy labels produced by the ASR system (see [18]), i.e., 13%/14% WER on *SOL-Cnt* and *UWB-ATCC* test sets.

Breaking the paradigm, acoustic or text-based speaker diarization? On challenging tasks such as ATC, where the rate of speech is high and contains mainly close-talk recordings, the standard acoustic-based SD systems are prone to fail and merge two or more segments together. An example is *SOL-Cnt* database (see Table 5.4) where $\sim$38% of the test set contains more than one speaker or/and segment per utterance (i.e., *'Mixed'*). We compare acoustic-based and BERT SD on private (*SOL-Cnt*) and public (*UWB-ATCC*) test sets. Similar to *SOL-Cnt*, *UWB-ATCC* set contains more than one speaker per utterance. We list the results in Table 5.6. In order to contrast both approaches, we compute the JER on the extracted segments, not on the text-level tokens (as done before). Both systems use the same SAD for segmentation. The acoustic-based SD, uses the Hungarian algorithm [274] for assigning the system clusters to the reference speakers. As a result, it evaluates SCD and clustering without identifying the speaker roles. For estimating the DER, we align the text with audio data and prepare the labeled segments from it. Using this alignment, the output of the BERT SD system is comparable to the acoustic-based diarization system. For computing the scores in all systems, the collar of 150 msec was considered. We found out that in noisy conditions, acoustic-based SD mistakenly oversplits the segments with one speaker (either ATCo or pilot). However, the BERT SD seems to be very robust on these segments (3.0/3.7% → 5.8/7.8% DER for ATCo/pilot of *SOL-Cnt* test set). Even in the mixed scenario of this set, the BERT SD system (9.5% DER) extended with data augmentation outperformed the acoustic-based model (10.3% DER) by 7.7%, relatively. On a cleaner set with shorter segments, VBx system shows the best performance on the segments with one speaker. However, in the mixed segments, the BERT SD system outperformed the VBx by a marginal improvement.

Conclusions In this work, we demonstrated that acoustic-based tasks such as speaker diarization can be enhanced or even replaced by natural language processing techniques. Even including challenging tasks such as SD for ATC communications. Additionally, we developed a simple and flexible data augmentation pipeline for ATC text data. To the authors' knowledge, this is the first time that a BERT-based SD could fully replace an acoustic-based SD in the field of ATC.

5.3 Benchmarking Multiple Spoken Language Understanding Representations

This section covers an exhaustive evaluation of different representations to address the intent classification problem in a Spoken Language Understanding (SLU) setup. We benchmark text, lattice and a multimodal approach to perform the SLU intent detection. Our work provides a comprehensive analysis of what could be the achievable performance of different SOTA SLU systems under different circumstances, e.g., automatically- *vs.* manually-generated transcripts. We evaluate the systems on the publicly available SLURP spoken language resource corpus. Our results indicate that using richer forms of Automatic Speech Recognition (ASR) outputs, namely word-consensus-networks, allows the SLU system to improve in comparison to the 1-best setup (5.5% relative improvement). However, crossmodal approaches, i.e., learning from acoustic and text embeddings, obtains performance similar to the oracle setup, a relative improvement of 17.8% over the 1-best configuration, being a recommended alternative to overcome the limitations of working with automatically generated transcripts.

Publication Note

The material presented in this section is adapted from the following publication:

- E. Villatoro-Tello, S. Madikeri, J. Zuluaga-Gomez, B. Sharma, S. S. Sarfjoo, I. Nigmatulina, P. Motlicek, A. V. Ivanov, and A. Ganapathiraju, “Effectiveness of text, acoustic, and lattice-based representations in spoken language understanding tasks,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2023, pp. 1–5

Supplementary materials related to this section:

- **Code - GitGub repository at:** https://github.com/idiap/slu_representations

Minor contributions Validation of the HERMIT architecture. Participated in the article write up and results analysis.

5.3.1 Introduction

Spoken Language Understanding (SLU) is the underlying key component of interactive smart devices such as voice assistants, social bots, and intelligent home devices. Effectively interpreting human interactions through classification of intent and slot filling plays a crucial role in SLU. Therefore, it is not surprising that the SLU problem has received substantial attention in industry and academia. A comprehensive and exhaustive introduction to NLU and SLU is in Section 2.2. An important aspect of current non-E2E SLU systems is that they rely on an ASR system for transcripts generation. Thus, there have been efforts to design tighter integration of ASR and NLU systems beyond 1-best ASR results, e.g., by means of encoding several ASR hypotheses through lattice-based representations. A lattice is a compact representation encoding multiple ASR hypotheses obtained at the decoding step. Its use has shown to be key in boosting the

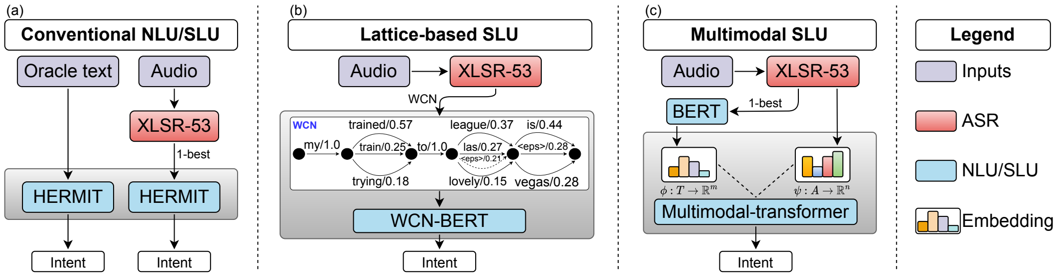


Figure 5.6: Overview of the considered NLU/SLU methodologies for the proposed experiments.

performance of IR systems [275].

In this direction, there are several works adopting word confusion networks (WCNs) as input to NLU systems to preserve information in possible hypotheses [276, 277]. The main advantage of WCN-based approaches is that they are less sensitive to the ASR errors. Finally, recent approaches based on multi-modal information have been proposed [84]. The main motivation behind this idea is founded on how humans interpret, in the real world, the meaning of an utterance and corresponding semantics from various cues, thus, assuming that the acoustic and linguistic content of a speech signal may carry complementary information for deriving robust semantic information of an utterance.

Overall, despite the promising results, there still exists a gap between the demonstrated capability of SLU systems and the requirements of an industrial application, e.g., a generalized voice assistant. For instance, E2E approaches mostly focus on databases with limited semantic complexity and structural diversity [278]. Additionally, most of the current benchmarks on SLU are widely saturated, where the obtained performances (F1-scores) are near perfect. Examples of such cases are results reported on ATIS [279], Fluent Speech Commands [82], or SNIPS [280] datasets. Hence, to validate the robustness of recent SLU approaches under a more realistic scenario, it becomes necessary to focus on SLU tasks that incorporate more complex semantics and numerous intent classes and slots. To the best of our knowledge, such benchmarking, comparing the wide variety of SLU approaches, has not been performed recently.

In this work, we present an extensive analysis of different SLU techniques, ranging from pure text-based alternatives to methods that are able to process richer forms of ASR outputs. Overall, our work has three salient features: *(1)* we evaluate and compare under the same circumstances four big families of SLU approaches, namely: text-based, lattice-based, multimodal, and end-to-end, *(2)* our performed evaluation is done considering a more realistic scenario, i.e., where no access to manual transcriptions exists, but instead, ASR transcripts are given as input to the SLU systems, and *(3)* we describe several inconsistencies found in the SLURP [281] dataset, one of the most challenging test beds for SLU systems.

Figure 5.6 depicts an overview of the considered SLU techniques in our experiments: (a) conventional or pipeline-oriented NLU/SLU approaches, (b) Lattice-based SLU architectures, and (c) multimodal (text+acoustic) architectures. Although not shown in the figure, we also report the performance of very recent E2E methods.

5.3.2 Multiple NLU/SLU Representations

Conventional NLU/SLU systems We selected the HERMIT architecture [282] as the representative approach for this category of systems. HERMIT, a **HiER**archical **M**ult**I**-**T**ask Natural Language Understanding architecture, was designed for effective semantic parsing of domain-independent user utterances, extracting meaning representations in terms of high-level intents and frame-like semantic structures. According to the authors, HERMIT stands out for being a cross-domain, multi-task architecture, capable of recognizing multiple intents in human-machine interactions. The central motivation behind the design of the HERMIT architecture is the modeling of the dependence among three tasks, namely, dialogue acts identification, intents, and slots. For this, the authors addressed the NLU problem using a seq2seq model employing BiLSTM encoders and self-attention mechanisms, followed by CRF tagging layers. HERMIT was validated in two large datasets with a high number of intent labels (58 to 68 classes), reporting a performance of F1=86%. We re-implemented HERMIT in PyTorch [224], with the following changes: we exchanged the encoder layer based on ELMO embeddings with a BERT [87] encoder, we replaced the BiLTSM encoders by GRU modules and used the AdamW optimizer. We evaluate the performance of our implementation of HERMIT when either, manual transcriptions (1-best) or ASR outputs extracted from the XLS-R model are used, see Figure 5.6.a).

Lattice-based SLU One main limitation of pipeline SLU systems is their sensitivity to the errors present in the ASR transcriptions. There are SLU systems robust against ASR errors based on lattices and WCNs [277, 283]. Although both, word lattices and WCNs contain more information than N-best lists, WCNs have been proven more efficient in terms of size and structure, thus representing a more plausible alternative when designing SLU systems that receive as input a graph-based structure. We re-implemented a very recent WCN-based approach, namely WCN-BERT [277]. Originally, the WCN-BERT architecture consists of three parts: a BERT encoder for jointly encoding, an utterance representation model, and an output layer for predicting semantic tuples. The BERT encoder exploits posterior probabilities of word candidates in WCNs to inject ASR confidences. Multi-head self-attention is applied over both WCNs and system acts to learn context-aware hidden states. The utterance representation model produces an utterance-level vector by aggregating final hidden vectors. Finally, WCN-BERT adds both discriminative and generative output layers to predict semantic tuples. WCN-BERT stands out for being able to leverage the timing information and confidence scores that are part of standard WCNs. Authors evaluated the performance of WCN-BERT on DSTC2 dataset [284], a corpus of dialogs in the restaurant search domain between a human and automatic agent (i.e., human-machine conversations) reporting and overall F1=87.91%. For our experiments, we only keep the WCN-BERT encoder and the multi-head attention layers to generate the utterance-level representation from the original model in [277]. On top of this, we concatenate a fully-connected layer to perform intent classification.

Multimodal SLU It refers to the process of embeddings alignment for explicitly minimizing the distance between speech embeddings and the text embeddings from state-of-the-art text encoders like BERT [87]. Thus, the speech embeddings that are used for downstream tasks are made to share a common embedding space with the textual embeddings, leading to better performance in SLU tasks, e.g., intent detection. However, there are a few challenges involved in the process of modeling such multimodal human language time-series, namely: 1) inherent non-aligned data due to variable sampling rates for the sequences from each modality; and 2) long-range dependencies between elements across modalities. In order to address these problems, we implemented a solution based on the Multimodal Transformer (MulT) [285]. MulT depicts an end-to-end model that extends the standard Transformer network [61] to learn representations directly from unaligned multimodal streams. At the heart of MulT, there is a cross-modal attention module, which attends to the crossmodal interactions at the scale of the entire utterances. It merges multimodal time-series via a feed-forward fusion process from multiple directional pairwise crossmodal transformers. Specifically, each crossmodal transformer serves to repeatedly reinforce a target modality with the low-level features from another source modality by learning the attention across the two modalities' features.

In our experiments, we adopted the ideas proposed in MulT [285]. Hence, given two input modalities, each crossmodal transformer block (one for each modality) keeps updating its sequence. Thus, the crossmodal transformer learns to correlate meaningful elements across modalities. As a final step, outputs are concatenated and passed through a self-attention module to collect temporal information to make predictions. The last elements of the sequence are passed through fully-connected layers to make the intent prediction.

5.3.3 Databases and Experimental Setup

ASR Module with XLSR-53 As shown in Figure 5.6, we consider the *XLSR-53* model as main ASR component [51]. XLSR-53 learns cross-lingual speech embeddings by pretraining a single generic model from raw waveform of speech in multiple languages. The structure of XLSR is similar to Wav2Vec 2.0 [49], which is trained using contrastive loss over masked latent speech representations and jointly learns a quantization of the latent embeddings shared across languages. We then fine-tune XLSR-53 model [51] with 390h of English data from AMI and Switchboard datasets using E2E-LFMMI loss function [286, 287] with biphone units [222, 288, 36]. A grapheme-based lexicon of size 1M was used, and the LM was trained with 34M utterances from publicly available English datasets including People's speech, Fisher, Switchboard, AMI, Wikitext103, and subsets of Common Crawl and Reddit datasets. To improve XLSR-53's generalization on English conversational speech, we fine-tuned it using 560h of untranscribed data crawled from YouTube.[15] On the YouTube data, we followed an incremental semi-supervised

[15] This subset was selected from conversational video calls in English language.

Table 5.7: SLURP statistics. **SLURP**$_O$: original, while **SLURP**$_F$ is a cleaner version of SLURP.

Statistics	**SLURP**$_O$	**SLURP**$_F$
Audio Files	72,277	50,568
$\hookrightarrow$ Close range	34,603	25,799
$\hookrightarrow$ Far range	37,674	24,769
Duration [hr]	58	37.2
Av. length [s]	2.9	2.6
Nb. of intents	48	47

learning approach with four iterations [139]. For decoding, we use the WFST decoder from Kaldi [100], with a beam width of 15.

SLURP database To perform our experiments we used the SLURP dataset [281], a publicly available multi-domain dataset for E2E-SLU, which is substantially bigger and more diverse than other SLU resources. SLURP is a collection of audio recordings of single-turn user interactions with a home assistant. Table 5.7 contains a few statistics about SLURP. During a manual analysis, we found many inconsistencies in SLURP annotations. Basically, we identified cases where the manual transcription does not correspond to what is being said in its corresponding audio file. Thus, we considered as erroneous those audio files for which the manual transcription did not match with the automatic transcription, or whose transcripts were inconsistent (i.e., not the same) in the corresponding metadata files. By following this approach, we detected that nearly 30% (20h) of the original data contains some type of inconsistency. We refer as SLURP$_F$ to the subset of SLURP without these inconsistent files (see Table 5.7).

Model Training Experiments from **EXP1** - **EXP7** correspond to the results of conventional NLU/SLU techniques. We employ the implementation of the HERMIT architecture [282]. As can be observed in Table 5.9, differences among these experiments are on the type of data used for training and evaluating the HERMIT model, i.e., combinations of either manual transcripts or 1-best ASR outputs. **EXP8** - **EXP9** correspond to the experiments done using WCN-based representations. Notice that for both set of experiments, i.e., conventional NLU and WCN-based, we used the XLS-R model, not-adapted and adapted to SLURP, to obtain the transcripts and WCNs respectively. Finally, **EXP10-EXP12** corresponds to the experiments done using the crossmodal transformer. Similarly, we evaluate the performance of this approach under circumstances where the XLS-R model is not adapted to the target domain (EXP10), and when adapted to SLURP (EXP11), and one last experiment using acoustic embeddings obtained from HuBERT (EXP12) model.[16]

[16] To generate the acoustic embeddings we followed the SpeechBrain [289] SLURP recipe: https://github.com/speechbrain/speechbrain/tree/develop/recipes/SLURP

5.3.4 Results & Discussion

Table 5.8: WER% on SLURP Test sets with the XLSR-53 English model before and after adaptation with SLURP$_F$ train subset.

System	Dev (WER%)		Test (WER%)	
	Headset	All	Headset	all
No adaptation	23.4	34.0	23.0	34.4
Adapted to SLURP	13.3	16.1	13.0	15.5

Table 5.9 shows the obtained experimental results for all the benchmarked architectures. Column "Exp" indicates the name of the experiment, "Input Type" describes what type of data was used for training and evaluating (dev and test) the corresponding experiment. For those experiments with the tag *manual* it means ground truth transcriptions were used, while *1-best* refers to the automatically generated transcriptions using the XLSR-53 model. Column "XLSR-53 adaptation" indicates whether the XLSR-53 model was fine-tuned to the SLURP dataset. In order to do the XLSR-53 adaptation, the English ASR model described in Section 2.4 was fine-tuned with the train subset of the SLURP$_F$ data without changing the LM. ASR performances before and after fine-tuning to SLURP are given in Table 5.8. And finally, SLURP$_O$ and SLURP$_F$ depict what version of the SLURP dataset was used.

Notice that the results obtained in the test partition from the cleaned version of the data, i.e., SLURP$_F$, are usually better than those obtained in the original version of the SLURP dataset. To some extent, this is an indicator that the identified inconsistencies in the original SLURP dataset were affecting the benchmarked models, resulting in a miss classification of some intents types.

Experiments EXP1, EXP3, and EXP6 represent artificial scenarios, as in a real-world application we do not expect to have manual transcripts for test partitions. Nevertheless, the best performance under this configuration, e.g., F1=87% in the SLURP$_F$ for EXP6, represents an upper bound value. Interestingly, this value is even better than the performance obtained in the EXP1, i.e., considering only ground truth data. This may be due to an (unexpected) regularization effect caused by the noise contained in the 1-best transcripts from all the audios of the SLURP. Thus, it becomes really relevant that WCN-based approaches (EXP8 & EXP9) are able to obtain a competitive performance against the pipeline 1-best $\rightarrow$ 1-best NLU experiments (EXP4). Even though the not-adapted WCN model (EXP8) does not outperform EXP4, this result validates the existence and the impact of richer ASR hypotheses in the lattice, which helps improve the performance of the SLU system, especially in noisy data (SLURP$_O$).

Although WCN experiment EXP9 showed a good improvement against the 1-best scenario, multimodal experiments obtain a remarkable performance, comparable to the performance of the oracle experiment (EXP1). The main difference between EXP11 and EXP12 is that the former uses XLSR-53 adapted to SLUPR, while the latter fine-tunes HuBERT toward intent and slot classification in SLURP. This is the explanation for why EXP12 performance is slightly better than EXP11 (XLSR-53 adapted). Finally, as reference results from an E2E approach, the

Table 5.9: Accuracy (ACC) and F1-scores (F1) on intent classification for different representations. We test our approach with either manual or 1-best approaches. Manual refers to ground truth evaluation, while 1-best is obtained by using our ASR module XLSR-53. Thus, manual → manual represents the oracle scenario (upper bound), while 1-best → 1-best depicts a more real-world case scenario.

Exp.	**Input Type**	**XLSR-53**	**SLURP$_O$**				**SLURP$_F$**			
			Dev (↑)		**Test (↑)**		**Dev (↑)**		**Test (↑)**	
	train→dev-test	**adaptation**	ACC	F1	ACC	F1	ACC	F1	ACC	F1
Conventional NLU/SLU										
EXP1	manual → manual	*NA*	**0.89**	**0.88**	0.85	0.84	**0.88**	**0.87**	0.82	0.82
EXP2	manual → 1-best	✗	0.70	0.65	0.69	0.65	0.74	0.69	0.71	0.67
EXP3	1-best → manual	✗	0.85	0.85	**0.86**	**0.85**	0.86	0.86	**0.85**	**0.83**
EXP4	1-best → 1-best	✗	0.72	0.68	0.73	0.69	0.76	0.71	0.77	0.73
EXP5	manual → 1-best	✓	0.82	0.81	0.80	0.79	0.84	0.82	0.86	0.84
EXP6	1-best → manual	✓	**0.88**	**0.87**	**0.87**	**0.86**	**0.88**	**0.87**	**0.88**	**0.87**
EXP7	1-best → 1-best	✓	0.84	0.83	0.83	0.83	0.85	0.84	0.85	0.84
Lattice-based SLU										
EXP8	WCN	✗	0.68	0.67	0.68	0.68	0.69	0.68	0.68	0.68
EXP9	WCN	✓	**0.78**	**0.77**	**0.79**	**0.79**	**0.80**	**0.80**	**0.78**	**0.77**
Multimodal SLU										
EXP10	multimodal	✗	0.75	0.75	0.74	0.73	0.75	0.74	0.76	0.76
EXP11	multimodal	✓	0.82	0.82	0.83	0.83	0.83	0.83	0.82	0.82
EXP12	multimodal (HuBERT [205])	✓	**0.87**	**0.88**	**0.84**	**0.84**	**0.88**	**0.88**	**0.86**	**0.86**

SLURP recipe reports F1 values of F1$= 0.77$ and F1$= 0.88$ under configurations referred to as *direct* and *direct Hubert* respectively. More details can be found in the respective repository.[3] Overall, using a multi-modal approach seems to be the recommended option, as it guarantees the best performance. Although it should be considered that it represents a costly solution in terms of computational power. On the contrary, if access to manual transcripts is guaranteed for training an NLU/SLU system, independently of having (or not) the possibility to adapt the ASR model toward the target domain, the recommended solution would be to follow a traditional NLU pipeline.

Conclusions In this work, we successfully benchmark several neural architectures to perform NLU, pipeline SLU and multi-modal SLU. Our analysis includes SOTA NLU/SLU techniques and compares them in more realistic scenarios. The presented analysis shed light on state-of-the-art architectures in the SLU domain, helping future researchers to define more clearly the application scenario of their proposed solutions. As an additional contribution, we also put together a cleaner version of the well-known SLURP dataset. During our experimentation process, we found many inconsistencies between the manual annotations and what was really spoken in the audio files. This rise concern on the SLU field, as several papers have already reported results on this dataset without being aware of it.

6 Joint Speech Recognition and Spoken Language Understanding

Introduction

In preceding chapters, we investigated how to build ASR and SLU systems for challenging applications, such as ATC. In this chapter, we showcase the integration and optimization of multiple tasks within a single model through end-to-end training. In Section 6.1, we demonstrate how a unified encoder-decoder model can handle ASR, speech-to-text translation, cross-talk, and speaker change detection via special tokens. Later in Section 6.2, we extend similar methodologies to the XLSR-Transducer architecture, offering promising applications in streaming industrial scenarios. An illustrative overview of this chapter is provided in Figure 6.1.

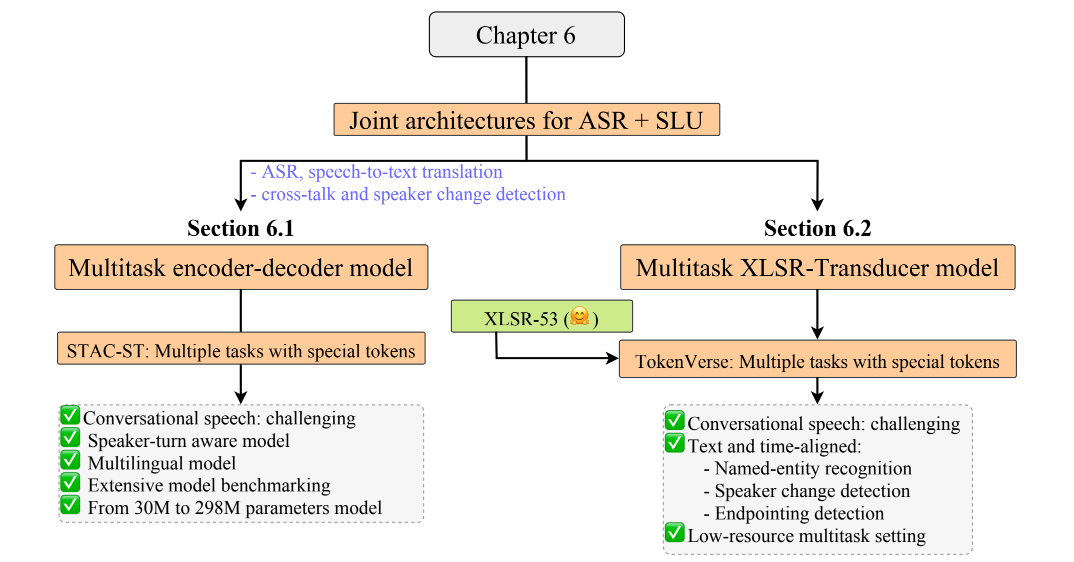

Chapter 6
Joint architectures for ASR + SLU
- ASR, speech-to-text translation
- cross-talk and speaker change detection
Section 6.1
Multitask encoder-decoder model
STAC-ST: Multiple tasks with special tokens
✓ Conversational speech: challenging
✓ Speaker-turn aware model
✓ Multilingual model
✓ Extensive model benchmarking
✓ From 30M to 298M parameters model
Section 6.2
Multitask XLSR-Transducer model
XLSR-53 (🤗)
TokenVerse: Multiple tasks with special tokens
✓ Conversational speech: challenging
✓ Text and time-aligned:
- Named-entity recognition
- Speaker change detection
- Endpointing detection
✓ Low-resource multitask setting

Figure 6.1: Overview of Chapter 6.

6.1 Token-Based Multitasking for Encoder-Decoder Models

Conventional speech-to-text translation (ST) systems are trained on single-speaker utterances, and they may not generalize to real-life scenarios where the audio contains conversations by multiple speakers. In this work, we tackle single-channel multi-speaker conversational ST with an end-to-end and multi-task training model, named Speaker-Turn Aware Conversational Speech Translation, that combines automatic speech recognition, speech translation and speaker turn detection using special tokens in a serialized labeling format. We run experiments on the Fisher-Callhome corpus, which we adapted by merging the two single-speaker channels into one multi-speaker channel, thus representing the more realistic and challenging scenario where multi-speaker turns and cross-talks occur. Experimental results across single- and multi-speaker conditions and against conventional ST systems, show that our model outperforms the reference systems on the multi-speaker condition, while attaining comparable performance on the single-speaker condition. We release scripts for data processing and model training.

Publication Note

This work was conducted during an internship at Amazon Web Services, Amazon in 2023.
The material presented in this section is adapted from the following publication:

- J. Zuluaga-Gomez, Z. Huang, X. Niu, R. Paturi, S. Srinivasan, P. Mathur, B. Thompson, and M. Federico, "End-to-End Single-Channel Speaker-Turn Aware Conversational Speech Translation," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 7255–7274

Supplementary materials related to this section:

- **Code - GitGub repository at:** https://github.com/amazon-science/stac-speech-translation

Major contributions Problem definition and experimental design and setup. Data preparation in the multi-task format for model training. Trained and fine-tuned the models for the experiments. Lead the work, including the paper write up.

6.1.1 Introduction

Speech translation (ST) has seen wide adoption in commercial products and the research community [290, 291] due to its effectiveness in bridging language barriers. ST aims to translate audio of source languages into text of the target languages. This problem was tackled by a cascaded approach that pipelines Automatic Speech Recognition (ASR) and Machine Translation (MT) over the last few decades [292, 293, 294]. However, end-to-end speech translation (E2E-ST) systems [295, 296] have recently gained increasing interest and popularity thanks to their simple architecture, less error propagation [297], efficient training process, and competitive performance [298].

Despite the significant advances in E2E-ST [299, 300], most ST systems to date have focused on translating isolated speech utterances from monologue speech [301], read speech [302] or

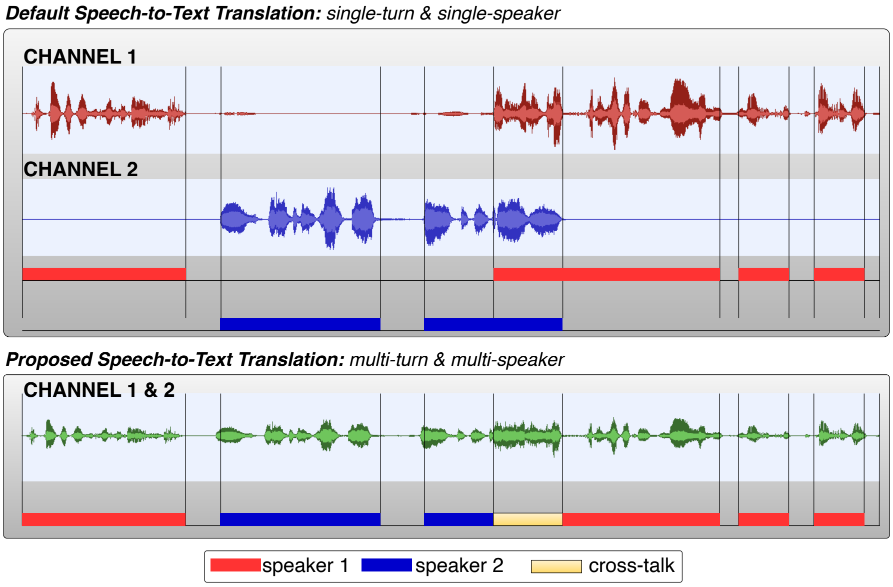

Default Speech-to-Text Translation: single-turn & single-speaker
CHANNEL 1
CHANNEL 2
Proposed Speech-to-Text Translation: multi-turn & multi-speaker
CHANNEL 1 & 2
speaker 1
speaker 2
cross-talk

Figure 6.2: A two-speaker multi-turn conversational segment. Previous work focuses on separated channels (top box) without considering cross-talks and speaker-turns. `STAC-ST` targets a more challenging scenario where multiple speakers converse, with occasional cross-talks due to merged channels (bottom box).

prompted speech [2]. Being trained on single-turn utterances, these systems may lack the ability to handle real-life scenarios in which multiple speakers converse, and sometime overlap, in the same audio channel [303].

In this work, we tackle the more challenging task of multi-speaker conversational ST. We refer to it as *multi-turn & multi-speaker* (MT-MS), as opposed to single-turn, which most ST systems implicitly assume. This is illustrated in Figure 6.2, where a “conversation" between two speakers recorded with separate channels (top) becomes more difficult to translate if the channels are merged (bottom), due to the appearance of speaker-turns and cross-talks. In particular, ST with cross-talks and speaker-turns is difficult because speech content of different sentences is mixed up or switched. While MT-MS speech has been studied in ASR [304], to the best of our knowledge, this is the first work that investigates it in end-to-end ST.

We tackle MT-MS ST with an approach we named **S**peaker-**T**urn **A**ware **C**onversational **S**peech **T**ranslation (`STAC-ST`). `STAC-ST` is a multi-task training framework that combines ASR, ST and speaker-turn detection using special tokens in a serialized labeling format. It is inspired by a recent speech foundation model, Whisper [188], which jointly trains ASR, X-to-English ST, voice activity detection, and language identification with 680kh of speech data using labeling-based multi-task learning. Our contributions are five-fold:

1. We introduce the task of multi-turn & multi-speaker ST, including cross-talks and speaker-turns, that expands the realm of ST which so far was limited to single-speaker utterances.
2. Our end-to-end `STAC-ST` model achieves state-of-the-art BLEU scores on Fisher &

CALLHOME, a corpus that allows to targets MT-MS, with no degradation on single-turn ST.
3. We explore a zero-shot scenario where MT-MS ST data is not available for training. `STAC-ST` improves ST up to 8 BLEU by leveraging MT-MS ASR targets, mitigating the necessity of parallel data, which is lacking within the community.
4. Besides serializing transcripts and translations at cross-talks, the `STAC-ST` model also shows to learn the task of time-aligned speaker change detection.
5. We conduct extensive ablation studies on important aspects of `STAC-ST`, including joint modeling of ASR & ST, impact of model size (up to 300M parameters), data size, and integration of task tokens. Thus, we shed light on the best practices for building conversational MT-MS ST systems.

Conversational Speech Translation Work on conversational ST [98, 305, 306] has mainly focused on single-speaker speech, either segmented manually or automatically, via voice activity detection. Manual segmentation was assumed in recent studies, based on Fisher and CALLHOME corpora, on cascaded ST [98], E2E-ST [296, 307], simultaneous ASR & ST [308], streamed ST [309], and multilingual ST [298]. Automatic segmentation was instead deployed with the MSLT corpus [1] to target streamed ST [310] as well as language-agnostic streamed ST [300].

In this work, we report results on the Fisher-CALLHOME corpus [303] which, similarly to the MSLT corpus, offers the opportunity to run contrasting experiments of single-speaker ST versus MT-MS ST, both without reference segmentation.

Speaker-Turn and Cross-Talk in ASR Speaker-turns and cross-talks have been already explored in the ASR field and commonly termed, multi-talker ASR. [105] proposes a serialized output training (SOT) strategy for multi-speaker overlapped speech recognition with special tokens. At inference time, word, and speaker tags are output in a serialized manner for an unlimited number of speakers. SOT was later ported to the streaming scenario in [311]. However, previous work argue that SOT may produce frequent speaker changes, which can degrade the overall performance. Thus, authors in [107] propose to explicitly incorporate boundary knowledge with a separate block for speaker change detection task and boundary constraint loss. Furthermore, multi-talker ASR has also been explored in the streaming [304] and non-streaming setups [312]. Another branch targets cross-talk & multi-talker ASR [313] using speech separation of long-form conversational speech [314] but these techniques have difficulty handling variable number of speakers and are not optimized end-to-end for ASR improvements. However, how to effectively deal with multi-speaker conversational ST has been neglected.

6.1.2 Speaker-Turn Aware Conversational Speech Translation System

This section describes our end-to-end multi-task learning model for multi-turn multi-speaker conversational ST.

System Diagram Figure 6.3 illustrates the proposed multi-task learning framework for MT-MS ST. The model is an encoder-decoder Transformer architecture inspired by [61]. The multitask training format using special tokens was inspired by Whisper [188], while the integration of CTC loss was inspired by [70].

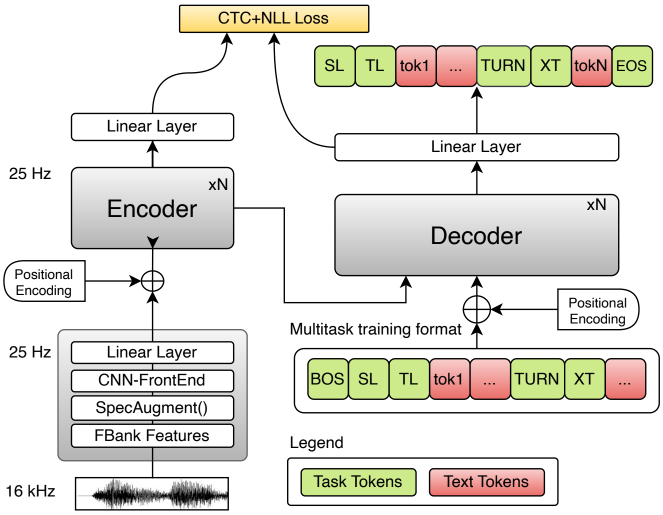


Figure 6.3: Proposed model architecture of STAC-ST for multi-turn & multi-speaker ST.

STAC-ST has a standard front-end module. First, frame-level 80-dimensional filterbank features are extracted from the audio[1] every 40ms. Second, we apply SpecAugment [160] on the input audio features, an effective data augmentation technique that masks out certain regions of the input filterbank features. Then, the audio augmented features are passed to a 2-layer CNN that outputs a 5120-dim vector (flattened 2D→1D output tensor from the CNN layer). Finally, this vector feeds a linear layer that generates the input to the encoder model. The decoder takes the encoder outputs and generates a sequence of text. Formally, for each speech segment, the filterbank features can be represented as: $X = \{\mathbf{x}_t \in \mathbb{R}^F\}_{t=1}^T$ and the reference transcription or translation as: $Y = \{w_n \in V\}_{n=1}^N$. Where, F is the feature dimension, T is the number of speech frames, N is the number of text tokens, and V is the vocabulary. During training of STAC-ST, we concatenate independent datasets $D_{ASR} = (X, Y_{ASR})$ and $D_{ST} = (X, Y_{ST})$, for ASR & ST, respectively. Samples of training mini-batches are jointly drawn from D_{ASR} and D_{ST}.

Serialized Labeling Based on Task Tokens A key component of the model is the serialized multi-task labeling framework based on special tokens. As shown in Figure 6.3, besides the text tokens, special tokens are used to specify the task. There are four types of task tokens, i.e., [SL] (source language), [TL] (target language), [TURN] (speaker-turn), and [XT] (cross-talk). The first two tokens are language tokens that define the task for either ST (when [SL] $\neq$ [TL]) or ASR (when [SL] $=$ [TL]). [TURN] and [XT] specify the auxiliary tasks of detecting speaker-turn changes and cross-talks, which are critical for MT-MS speech processing. The latter two are more aligned to acoustic tasks. Note that cross-talks always occur during speaker-turn changes, so [XT] always follows [TURN].

We concatenate transcripts or translations sequentially, appending [TURN] and [XT] tokens when needed. If utterances u_t and u_{t+1} overlap in time, we append the targets of utterance u_{t+1}

[1] The audio is always down- or up-sampled to 16 kHz.

after utterance u_t. The order of utterances is determined by their start time. At training time, we prepend `[SL]` and `[TL]` tokens to each sample of D_{ASR} and D_{ST}, while at inference, both are preset to specify the desired task.

Joint CTC and NLL Loss `STAC-ST` jointly models ASR and ST by balancing CTC [45] and Negative Log-Likelihood (NLL) losses [315], according to:

$$\mathcal{L} = \lambda \cdot \mathcal{L}_{CTC}(Y|X) + (1 - \lambda) \cdot \mathcal{L}_{NLL}(Y|X), \tag{6.1}$$

$\mathcal{L}_{CTC}$ and $\mathcal{L}_{NLL}$ are computed by appending linear layers with dimension V on top of the encoder and decoder, respectively. Figure 6.3 shows the proposed joint CTC/NLL loss training scheme [70]. In practice, the CTC loss models a probabilistic distribution by marginalizing over all possible mappings between the input (audio features, sampled at 40 ms) and output sequence (transcription or translation). We refer readers to the original implementation in [45], for more details. Moreover, CTC loss has been proven to aid ST by helping to stabilize encoder representations at early stages of training, i.e., allowing the decoder to learn soft alignment patterns faster [72]. Note that we do not include language tokens, `[SL]` and `[TL]`, for $\mathcal{L}_{CTC}$ computation because they do not correspond to acoustic features. Following previous work [316, 317], we set the weight λ of the CTC loss to 0.3.

This section introduces the datasets and metrics we used for evaluation, as well as architecture and training details of `STAC-ST`.

6.1.3 Databases and Experimental Setup

Conversational Multi-Turn & Multi-Speaker ST We use the Fisher and CALLHOME corpora which respectively comprises 170 hr and 20 hr of audio and transcripts of telephone conversations in Spanish.[2] The Spanish-to-English translations are available from [303]. We refer to them as Fisher-CALLHOME and summarize the data statistics in Table 6.1. This corpus is well suited for MT-MS ST, as it contains a significant amount of labeled data and non-segmented (audio) long conversation between speakers. We merged Fisher and CALLHOME for training and up-sampled the audio to 16 kHz.

Table 6.1: Fisher-CALLHOME corpus statistics.

	Fisher				**CALLHOME**		
Statistics	train	dev	dev2	test	train	dev	test
Duration [h]	150	4.0	3.9	4.0	16	4.0	2.7
#Utterance [k]	138	3.9	3.9	3.6	15	3.9	1.8
Speech act. [%]	97	97	98	98	78	80	58

Segmentation Each conversation on Fisher-CALLHOME occurred between two speakers with multiple turns over two channels (one speaker per channel). For MT-MS ST experiments,

[2] LDC2010S01, LDC2010T04, LDC96S35, LDC96T17

we merge the two channels into one, which creates natural speaker changes and cross-talks as illustrated in Figure 6.2. Human annotations in Fisher-CALLHOME provide time-aligned audio utterances, transcripts and translations, and have been used to segment each channel into single-turn utterances in prior work [e.g., 298]. Figure 6.4 plots the distributions of segment duration in the corpus. We observe that the majority of single-turn segments are less than 5 seconds long. To build models with manageable size and computation, following [188], we segment the merged-channel conversations into chunks of up to 30 seconds. For this step, we first used an off-the-shelf VAD-based segmentation tool, SHAS [318], but we realised that the resulting duration histogram is almost uniform and far from the natural segmentation. Hence, we decided to rely on the manual time annotations as follows. Starting from the first utterance $start$, we find the farthest utterance, end such that $end - start$ is up to 30 seconds. We extract audio within this span as one segment and repeat this procedure until the last utterance end is reached. Note that one segment may stretch over multiple utterance $start$ and end, so it may include silences, noise, speaker changes and cross-talks. We use this as the primary MT-MS segmentation strategy throughout this section unless otherwise stated.

Additional ASR & ST Corpora Fisher-CALLHOME has limited training data size, so we explore additional corpora to improve our model and to evaluate its generalization ability. We also use the official CoVoST2 [2] splits for Spanish-English ST (156 hr) and Common Voice (CV) [3] [97] splits for Spanish ASR (458 hr) as additional training data. Even though these corpora are not in conversation domain, they may still help speech modeling in general.

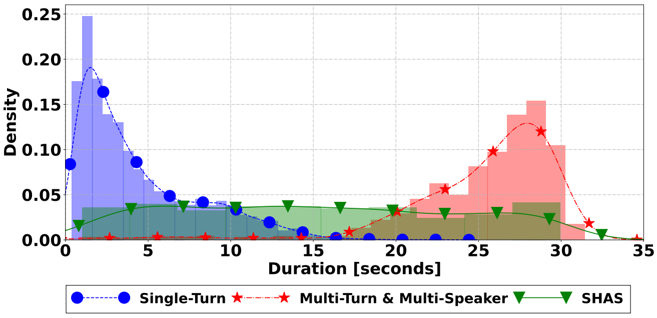


Figure 6.4: Fisher-CALLHOME test set distribution of segment length with three different segmentation approaches: single-turn, MT-MS, and SHAS.

CoVoST2 and CV corpora are composed of single-turn pre-segmented utterances. To generate data consistent with our MT-MS segmentation, we randomly concatenate audio utterances and yield segments of up to 30 seconds. Note that these synthetic MT-MS segments contain no silences and cross-talks, but still have speaker-turn changes (labeled by `[TURN]`).

Evaluation Metrics We report case-insensitive BLEU using SacreBLEU[4] [319] for translation and Word Error Rate (WER) for ASR. Note that we (1) remove all special task tokens before

[3] Version: `cv-corpus-13.0-2023-03-09`.

[4] Signature: `nrefs:N|case:lc|eff:no|tok:13a|smooth: exp|version:2.3.1`. (Fisher N=4 and CALLHOME N=1).

computing each metric and (2) evaluate on MT-MS segmentation unless otherwise stated.

Hyper-Parameters We experiment with three model sizes, S(mall), M(edium), and L(arge), with increasing dimension (256, 512, 1024), number of encoder layers (12, 14, 16), number of heads (4, 8, 16), with same number of decoder layers (6) and FFN dimension set to 4x the model dimension. Their numbers of parameters are 21M, 86M, and 298M, respectively. We use the S-size model by default and scale up to larger sizes when out-of-domain training data are added. We apply BPE sub-words [67] on both translations and transcripts with 5K operations. We create a joint BPE model for the language pair, or when we add CV+CoVoST2 corpora.

We train for 100k steps the S-size models and 200k steps the M- and L-size models. We use AdamW [186] optimizer with a peak learning rate of $5e^{-3}$ for the S model and $1e^{-3}$ for M and L models. The learning rate scheduler has warmup and cooldown phases, both taking 10% of the total training steps [320]. We set dropout [157] to 0.1 for the attention and hidden layers, and use GELU (Gaussian Error Linear Units) as the activation function [158]. We use gradient norm clipping [258][5] and SpecAugment [160] for data augmentation. The training configuration and architecture are based on a LibriSpeech recipe for Transformer-based ASR from the SpeechBrain toolkit [289].[6]

6.1.4 Results & Discussion

Our experimental results document three properties of the `STAC-ST` model: (1) robustness to the MT-MS ST condition with no degradation in the single-turn ST condition; (2) ability to leverage speaker-turn and cross-talk information, which translates into improved WER and BLEU scores; (3) ability to perform time-aligned speaker change detection.

Table 6.2: ASR and ST performance of `STAC-ST` with different training data configurations. Joint training with single-turn and multi-turn data of both ASR and ST tasks achieves the best scores.

Training data configuration				**Fisher**		**CALLHOME**	
Single-Turn		Multi-Turn		WER	BLEU	WER	BLEU
ASR	ST	ASR	ST	(↓)	(↑)	(↓)	(↑)
1)	✓		✓	-	28.3	-	8.5
2)		✓	✓	29.4	41.5	49.9	14.7
3) ✓	✓	✓	✓	**25.8**	**46.8**	**42.1**	**17.9**
4) ✓	✓			40.2	29.3	57.9	8.9
5) ✓	✓	✓		**25.8**	35.6	42.3	11.7
6) ✓	✓		✓	44.9	43.7	68.2	15.5

Multi-Task Learning We explored various training data configurations for multi-task learning (see Table 6.2). We started with training a model with only MT-MS data. The training failed to converge, because combining single-turn utterances to create longer (max 30s) MT-MS segments greatly reduces the number of training samples. We tackled this issue by

[5] $max\_grad\_norm = 5.0$.

[6] https://github.com/speechbrain/speechbrain/tree/develop/recipes/LibriSpeech/ASR/transformer

augmenting the training data with auxiliary tasks, as explained below and reported in Table 6.2.

Joint training of single-turn and multi-turn tasks is beneficial Adding single-turn ST data shows to stabilize training (Row-1 of Table 6.2). Using both single-turn and multi-turn ST+ASR data further improves ST quality (Row-3). Although single-turn and multi-turn data share the same utterances, split/concatenation-based data augmentation is known to be effective in the low-resource training regime [321, 322].

Joint training of ST and ASR is beneficial Just adding multi-turn ASR data also stabilizes the training (Row-2). By adding both single-turn and multi-turn ASR data for joint training on top of Row-1, both BLEU and WER are improved by a significant margin (Row-3).

Multi-turn ASR data help multi-turn ST In our training data, there are more labeled single-turn ST data and multi-turn ASR data than multi-turn ST data. We tested a zero-shot setting where, for the multi-turn condition, is only covered by ASR training data (Row-5). Comparing to training with single-turn ST+ASR data only (Row-4), the resulting model brings 3-8 BLEU gains. We hypothesize that, as the encoder is target-language-agnostic, the acoustic representations and the turn detection capacity learned from multi-turn ASR data does partially transfer to the ST task.

Table 6.3: ASR and ST performance of `STAC-ST` with the incremental addition of task tokens. Modeling speaker-turn and cross-talk detection with `[TURN]` and `[XT]` tokens enhances ASR and MT accuracy.

	Fisher		CALLHOME	
Task tokens	WER↓	BLEU↑	WER↓	BLEU↑
`[SL]`, `[TL]`	26.4	45.0	43.7	16.6
+ `[TURN]`	**25.8**	45.2	43.1	17.6
+ `[XT]`	**25.8**	**46.8**	**42.1**	**17.9**

On the contrary, multi-turn ST does not seem to help multi-turn ASR, as shown by comparing WER scores in Row-4 and Row-6. We hypothesize that the non-monotonicity of the multi-turn ST task disrupts multi-turn ASR performance [72]. However, this can be fixed by adding back multi-turn ASR data (Row-3). Note that we will use the Row-3 data configuration for the rest of this work.

Speaker-Turn and Cross-Talk Detection The `STAC-ST` multi-task learning framework also encodes speaker-turn and cross-talk information with task tokens `[TURN]` and `[XT]`. We run experiments to study how these task labels impact on ASR and ST performance in MT-MS setting and how they even enable speaker change detection.

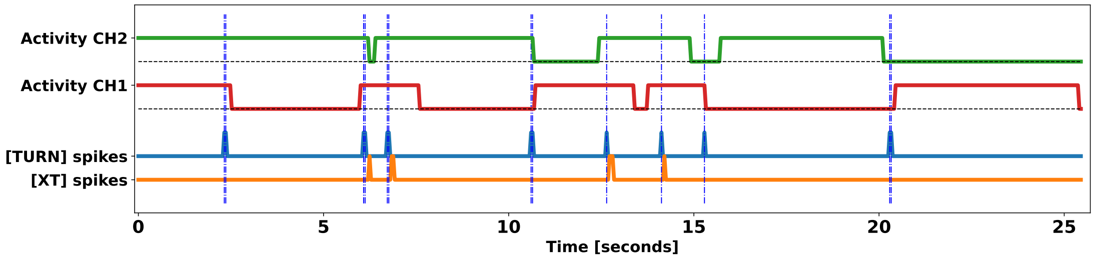


Figure 6.5: Speaker activity on a Fisher corpus sample. On the top, ground truth human annotation on two audio channels. On the bottom, CTC spikes of turn and cross-talk tokens detected by STAC-ST in the merged channel.

Modeling speaker-turn and cross-talk detection helps multi-speaker ST and ASR We run experiments by ablating the two task tokens. Evaluation results in Table 6.3 show that incrementally adding speaker-turn and cross-talk detection tasks improves translation and transcription quality measured by BLEU and WER. These results support the hypothesis that explicitly learning the two tasks helps the model to better handle MT-MS scenarios.

Table 6.4: Speaker change detection performance measured by F1, MDR and FAR. We compare STAC-ST with a well-known speaker diarization toolkit, PyAnnote. The strongest L-size STAC-ST model (from Table 6.5) shows on-par F1-score with PyAnnote. Tolerance is set to 0.25s.

	Fisher			CALLHOME		
System	F1↑	MDR↓	FAR↓	F1↑	MDR↓	FAR↓
PyAnnote	75.8	**26.8**	21.4	81.2	**20.9**	15.0
STAC-ST	74.9	31.3	17.7	80.6	25.6	**12.1**
STAC-ST (L)	**77.6**	28.6	**15.0**	**81.3**	23.5	13.2

Modeling speaker-turn and cross-talk detection enables the model to perform speaker change detection The CTC loss helps the encoder to align input audio to text tokens per acoustic frame, including the two task tokens. We trace speaker-turns and cross-talks in the timeline by (1) first running a forward pass on the encoder to extract audio-text temporal alignments and then we (2) locate the spikes of the liner layer on top of the encoder (aka. CTC spikes) only for [TURN] and [XT] tokens. As illustrated in Figure 6.5, the CTC spikes align remarkably well with actual edges of speaker activities. By leveraging available annotations in Fisher-CALLHOME test sets, we measure speaker change detection performance with three standard metrics: False Alarm Rate (FAR), Miss Detection Rate (MDR) and F1-score. The FAR computes the rate at which STAC-ST outputs a [TURN] CTC spike when there is no speaker change. The MDR computed the rate that STAC-ST misses generating [TURN] tokens at speaker changes. While the former two are widely used in speaker segmentation research [323], the F1-score provides an overall assessment of the performance.

To compute these metrics, we first prepare Rich Transcription Time Marked (RTTM) files for each test set from the time-aligned CTC [TURN] spikes. We compared performance of two

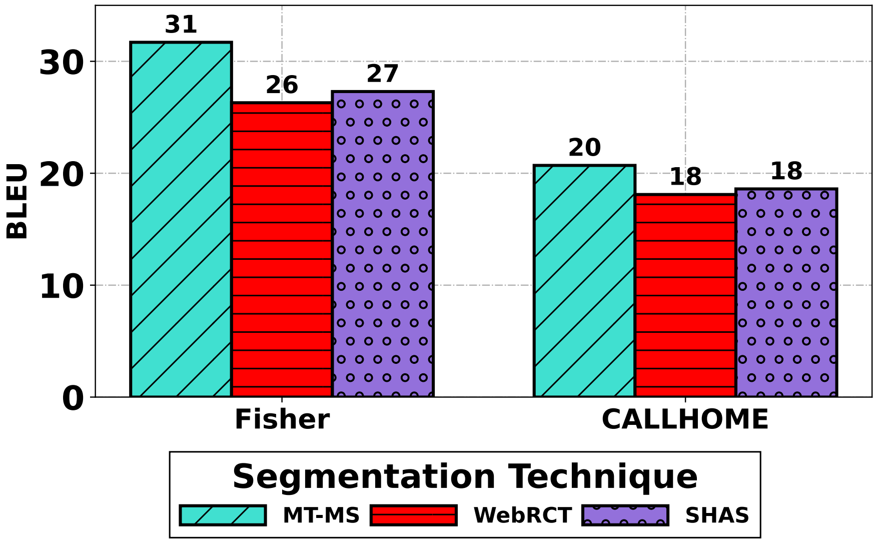


Figure 6.6: ST performance on Fisher-CALLHOME test data using different segmentation techniques for long-form audio: MT-MS (ours), WebRCT, and SHAS. BLEU scores of using VAD-based tools (either WebRCT or SHAS) for test data segmentation are below the ones using our MT-MS segmentation.

`STAC-ST` models (S and L) against a reference system, that is the popular speaker diarization pipeline of the PyAnnote toolkit [324]. From results listed in Table 6.4, `STAC-ST` gets on-par F1-score vs. the reference system in the Fisher-CALLHOME test sets. Also, we note that using a stronger `STAC-ST` model improves by 2.5 absolute the F1 score. These results corroborate the importance of the `[TURN]` task tokens for improving ASR and ST quality.

6.1.5 Benchmarking STAC-ST

We run extensive benchmarks to compare `STAC-ST` with related work in various settings, including (1) different audio segmentation strategies, (2) model size, and (3) evaluation on single-turn ST.

MT-MS vs. VAD Segmentation A common practice for translating long-form audio files is to first segment them into smaller chunks based on voice activity detection (VAD). We compare our MT-MS segmentation approach with two popular VAD-based audio segmenters, i.e., WebRCT [325] and SHAS [318], on the channel-merged Fisher-CALLHOME test sets.[7] When the audio and reference translation segments are not aligned, like in the case of VAD-based segmentation, the standard process is to first concatenate translation hypotheses and then align and re-segment the conversation-level translation based on the segmented reference translation.[8] However, our

[7] More details in Appendix A.1.

[8] `mwerSegmenter` [326] has been used in IWSLT [291, 290] for this purpose.

preliminary results show that this process yields poor BLEU scores, partially because VAD treats noise as speech, which leads to noisy translation and misalignment. Therefore, we calculate BLEU scores on concatenated hypotheses and references for the whole conversation. BLEU scores in this section are not comparable with the ones reported elsewhere.

As shown in Figure 6.6, for both Fisher and CALLHOME test sets, BLEU scores of using VAD-based tools (either WebRCT or SHAS) for test data segmentation are below the ones using our MT-MS segmentation. Despite being popular in conventional speech translation, segmenting long-form audio with VAD-based tools is not the best choice for handling multi-talks conversations with speaker-turns. Thus, we resort to using MT-MS segmentation based on human annotations for preparing the test data. This highlights the future work of producing robust segmentation on noisy long-form conversational audio.

Scaled `STAC-ST` vs. Whisper Given the lack of prior work on MT-MS ST, we compare `STAC-ST` against a strong multi-task model, i.e., Whisper [188]. Whisper is trained with over 2,000 times more speech data than our model (although Fisher-CALLHOME is not included among them) and its smallest version is larger than `STAC-ST` S. To fill part of the gap, we added more speech training data to `STAC-ST` with size M and L. Results in Table 6.5 demonstrate that when we add out-of-domain training data and scale the model accordingly [327, 78, 320], `STAC-ST` achieves better BLEU and WER scores than Whisper with comparable model sizes, although our training data is still three orders of magnitude smaller.

Table 6.5: ASR and ST performance with increasing model size of `STAC-ST` and Whisper. `STAC-ST` achieves better BLEU and WER scores than Whisper with comparable model sizes.

	Fisher		CALLHOME	
Model	WER↓	BLEU↑	WER↓	BLEU↑
Whisper-tiny (39M)	45.0	11.5	59.8	2.4
Whisper-base (74M)	36.7	29.0	49.2	8.4
Whisper-small (244M)	29.1	46.7	37.9	19.2
`STAC-ST` S (21M)	25.8	46.8	42.1	17.9
`STAC-ST` M (86M)	23.8	49.4	38.3	20.4
`STAC-ST` L (298M)	**23.5**	**50.0**	**38.5**	**21.0**

`STAC-ST` for Single-Turn ST To position `STAC-ST` against previous work on ST, we also run experiments under the conventional single-turn ST condition. These experiments permit from one side to see how our end-to-end multi-task learning approach performs on a specific input condition, and on the other side to compare `STAC-ST` against four previous models trained and evaluated on the same task. To allow for comparing results across single-turn and MS-MT conditions, we also report performance with three Whisper systems. Results of these experiments are reported in Table 6.6. We observe that all our `STAC-ST` models show to be competitive with the previous models, also optimized on the Fisher-CALLHOME task. Comparison against the Whisper models confirm the trend observed in Table 6.5 under the MS-MT condition. Overall,

Table 6.6: ASR and ST performance with the official single-speaker manual segmentation. Previous work results and Whisper baselines are provided. Our strongest model, STAC-ST L yields the best scores.

	Fisher		**CALLHOME**	
Model	WER↓	BLEU↑	WER↓	BLEU↑
Casc. ST [303]	36.5	-	65.3	11.6
Multi-task [296]	23.2	48.7	45.3	17.4
ESPnet [328]	**18.7**	50.5	37.6	21.7
E2E-ST [298]	22.9	46.3	44.5	17.2
Whisper-tiny (39M)	44.1	9.0	58.5	2.2
Whisper-base (74M)	34.8	25.4	48.7	6.5
Whisper-small (244M)	28.1	45.3	36.5	16.8
STAC-ST S (21M)	20.9	49.1	36.3	20.1
STAC-ST M (86M)	**18.9**	52.3	31.4	22.1
STAC-ST L (298M)	**18.8**	**52.6**	**31.0**	**22.4**

STAC-ST L yields the best BLEU scores on both Fisher and CALLHOME.

Conclusions In this work, we present STAC-ST, an end-to-end system designed for single-channel multi-turn & multi-speaker speech translation that uses a multi-task training framework to leverage both ASR and ST datasets. We demonstrate that STAC-ST generalizes to both standard pre-segmented ST benchmarks and multi-turn conversational ST, the latter being a more challenging scenario. STAC-ST also shows to learn the task of speaker change detection, which helps multi-speaker ST and ASR. We investigate different aspects of STAC-ST, including the impact of model and data size, automatic segmentation for long-form conversational ST, zero-shot multi-turn & multi-speaker ST without specific training data. Overall, this work sheds light on future work towards more robust conversational ST systems that can handle speaker-turns and cross-talks.

6.2 Token-Based Multitasking for Transducer Models

In traditional conversational intelligence from speech, a cascaded pipeline is used, involving tasks such as voice activity detection, diarization, transcription, and subsequent processing with different NLU/SLU models for tasks like semantic endpointing and named entity recognition (NER). This work introduces ***TokenVerse***, a single Transducer-based model designed to handle multiple tasks. This is achieved by integrating task-specific tokens into the reference text during ASR model training, streamlining the inference and eliminating the need for separate NLP models. In addition to ASR, we conduct experiments on 3 different tasks: speaker change detection, endpointing, and NER. Our experiments on a public and a private dataset show that the proposed method improves ASR by up to 7.7% in relative WER while outperforming the cascaded pipeline approach in individual task performance. Additionally, we present task transfer learning to a new task within an existing TokenVerse.

Publication Note

The material presented in this section is adapted from the following publication:

- S. Kumar, S. Madikeri, J. Zuluaga-Gomez, I. Nigmatulina, E. Villatoro-Tello, S. Burdisso, P. Motlicek, K. Pandia, and A. Ganapathiraju, "TokenVerse: Unifying Speech and NLP Tasks via Transducer-based ASR," in *arXiv:2407.04444*, 2024

Minor contributions Contributed to data preparation for TokenVerse, specifically, for the CallHome dataset. Participated in the article write up and results analysis.

6.2.1 Introduction

Automated analysis of conversational audios has a wide range of practical applications, including in contact center analytics [329, 109]. Traditionally, conversational audios are transcribed with intermediate voice activity detection (VAD) [330] or endpointing [331] and diarization [332]. Afterward, separate NLP pipelines are employed on the transcripts to perform tasks such as named entity recognition (NER) [333], among others, to comprehend the conversation's structure and content [108, 334]. Using separate models for each subtask (optimized independently) has drawbacks [335] such as error propagation and a potential mismatch between automatic speech recognition (ASR) metrics and the final task. For instance, the best ASR hypothesis may not be optimal for the final task. Moreover, the cascaded approaches could translate to increased compute and latency, which will be exacerbated by the introduction of a new task.

In this work, we introduce `TokenVerse`, a neural Transducer [53] model capable of learning ASR and multiple additional tasks through the incorporation of task tokens. In contrast to the multi-head based multitasking approaches explored in previous studies [336, 337, 338], `TokenVerse` distinguishes itself by generating tokens directly within the ASR hypothesis, as illustrated in Figure 6.7a. Leveraging the transducer architecture [53], we can attain text-audio

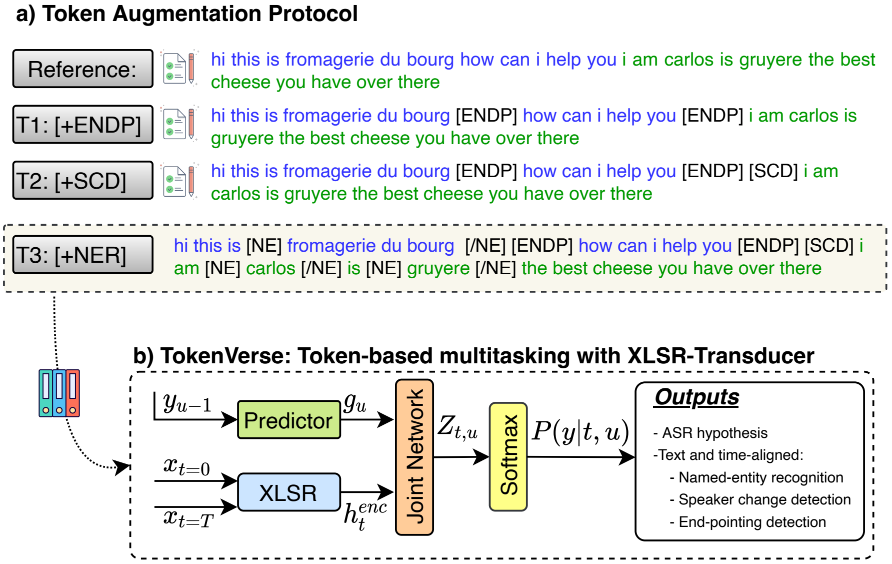

a) Token Augmentation Protocol
Reference:
hi this is fromagerie du bourg how can i help you i am carlos is gruyere the best cheese you have over there
T1: [+ENDP]
hi this is fromagerie du bourg [ENDP] how can i help you [ENDP] i am carlos is gruyere the best cheese you have over there
T2: [+SCD]
hi this is fromagerie du bourg [ENDP] how can i help you [ENDP] [SCD] i am carlos is gruyere the best cheese you have over there
T3: [+NER]
hi this is [NE] fromagerie du bourg [/NE] [ENDP] how can i help you [ENDP] [SCD] i am [NE] carlos [/NE] is [NE] gruyere [/NE] the best cheese you have over there
b) TokenVerse: Token-based multitasking with XLSR-Transducer
y_{u-1}
Predictor
g_u
$x_{t=0}$
$x_{t=T}$
XLSR
h_t^{enc}
Joint Network
$Z_{t,u}$
Softmax
$P(y|t,u)$
Outputs
- ASR hypothesis
-Text and time-aligned:
- Named-entity recognition
- Speaker change detection
- End-pointing detection

Figure 6.7: a) Proposed unified token augmentation protocol for SCD, ENDP, and NER. b) TokenVerse unifies multiple speech and NLP tasks (e.g., *T1+T2+T3*) in a single model within the neural Transducer framework.

alignment for each output token, including those designated as task tokens. For example, we can perform NER directly in the acoustic domain, presenting potential utility in scenarios such as audio de-identification [339]. To address challenges in low-resource settings, we use self-supervised (SSL) trained XLSR-53 [51] model as an encoder in the transducer setup, leading to the XLSR-Transducer [11] (Figure 6.7b), which is introduced formally in Section 4.2. Previous works aim at modeling several tasks directly from speech using special tokens [340, 341], or ASR with speaker change detection (SCD) [342, 343, 338], VAD [188], speech-to-text translation [8], or timestamps [344], NER [335, 345] and multi-speaker ASR [106, 311].

Token-based multitasking offers multiple benefits, e.g., it has a fixed number of parameters while all tasks are predicted with standard decoding without increased latency. However, NLP tasks like NER in conjunction with other tasks from audio domains have not received much attention in the literature. Therefore, we consider 3 additional tasks alongside ASR: SCD, endpointing and NER. These tasks are selected to represent both audio and NLP domains. SCD is an audio task [346]. Endpointing can be viewed as an NLP task when conducting semantic endpointing [347], or as an audio task [331]. NER is an NLP task [333, 335]. They serve as suitable benchmarks for evaluating our proposed method.

6.2.2 `TokenVerse`

Through `TokenVerse`, we aim to train a single model for ASR (main task), speaker change detection (SCD), endpointing, and named entity recognition (NER). This is achieved by augmenting the reference text, with task tokens that denote special events at the acoustic level. In the following sections, we discuss the annotation protocol, dataset preparation, details of our ASR model and ablation experiments.

Token Augmentation Protocol We introduce “tokens" for tasks apart from ASR: `[SCD]` (speaker change detection), `[NE]` and `[/NE]` (named entity recognition), and `[ENDP]` (endpointing). An illustrative example is depicted in Figure 6.7a. We insert `[SCD]` token during text concatenation if there is a speaker change from one segment to another within an utterance. The `[ENDP]` token is inserted at the end of a segment text, considered as a semantic endpoint from the conversational context perspective. Note that occurrence of `[ENDP]` will be a superset of `[SCD]` because a speaker change indicates the completion of the previous speaker’s sentence. For NER, we insert `[NE]` before the start of a named entity (NE) and `[/NE]` after it is concluded, since it can comprise multiple words.

Dataset Preparation Our work is focused on conversational audios, which are typically long in duration (avg 5 minutes) and can’t be directly used for ASR training due to high GPU memory requirements. The dataset provides audio-text transcripts together with timestamp information for every segment within the long-form audio. For each sample, we begin with the first segment $start$ and find the farthest segment end such that the duration is up to 20 seconds. Audios within this range are extracted as one utterance, and this procedure is repeated until the last segment is consumed. Note that an utterance may span over multiple segments, potentially containing silences, noise, speaker changes, endpoints and numerous named entities. Afterward, we concatenate the text corresponding to all segments within an utterance, inserting token at appropriate positions according to our tasks. This multi-task dataset preparation approach applies universally across all datasets used in our experiments.

Table 6.7: Datasets statistics with token metadata per subset for the public and private datasets.

Datasets metadata			**Token-based metadata [%]**				
subset	#utt/word	dur [h]	`[SCD]`	`[NE]`	`[ENDP]`	#NE	#uniq
DefinedAI dataset							
train	10k/359k	40	1.9	3.6	2.1	6.5k	2350
dev	559/20k	2.25	2.0	3.6	2.1	379	232
test	1.1k/42k	4.5	1.9	3.4	2.0	727	378
CallHome dataset							
train	2.7k/198k	13	6.3	2.9	8.7	2.8k	1414
dev	641/52k	3	7.2	3.0	10.4	779	466
test	339/23k	1.5	6.0	3.0	9.6	351	220

6.2.3 `TokenVerse` Training & Inference

`TokenVerse` Training We train the XLSR-Transducer model [11] on the multi-task data which consists of XLSR encoder, state-less predictor [55] and joint networks (linear layer). The model is trained with pruned transducer loss [60]. We utilize SentencePiece [68] tokenizer to train subwords from the training text [67]. It is important to note that the text includes task-specific tokens, and splitting them into multiple subwords may degrade their prediction accuracy because the entire sequence of subwords for a token must be predicted correctly to count it as a valid token prediction. Hence, we ensure that tokens are represented by a single subword during their training.[9]

`TokenVerse` Inference We generate hypothesis with beam search. From the hypothesis, we can extract and align the predicted task tokens in the time domain. Since NER consists of two tokens, we extract words between a matched pair of `[NE]` and `[/NE]`. We discard any unpaired tag from the hypothesis. To obtain timestamps for `[SCD]` or `[ENDP]`, we note the acoustic frame index for which these tokens are emitted and calculate time information, i.e., XLSR acoustic embeddings have a frame duration of 25ms and a stride of 20ms. Particularly for `[SCD]`, the time-level token prediction enables subsequent tasks, e.g., diarization [343].

6.2.4 Ablations within TokenVerse

We conduct ablation experiments to understand how including or excluding tasks affects other tasks in the `TokenVerse`. Note that ASR is our primary task and is always included.

Single task For each task, we retain only the tokens specific to that task in the multi-task dataset and train our XLSR-Transducer model. This helps eliminate any detractor tasks that may affect the performance of the task being evaluated and serves as a baseline in this work.

Leave-one-task-out We systematically exclude tokens corresponding to a single task from the multi-task dataset and proceed to train our ASR model. These experiments aims to examine how the removal of a task affects all other tasks, including ASR. This provides insights into whether we should retain or discard any task in `TokenVerse` for optimal performance on a given task.

Task-Transfer Learning In conventional multi-head multi-task architectures [336, 337], integrating a new task typically necessitates fine-tuning the model on the specific task while keeping the base encoder and other heads frozen. We explore the viability of this extension for `TokenVerse` by fine-tuning the model, derived from the removal of a task, specifically on the new task. Furthermore, we evaluate its impact on both existing tasks and the performance of the new task in comparison to the overall performance when all-tasks are included.

6.2.5 Task-Specific Baselines, Metrics & Evaluation Protocol

In this section, we describe strong independent baselines for each task considered in this work.

[9] https://github.com/google/sentencepiece

Automatic Speech Recognition We train our XLSR-Transducer model [11] after removing all task tokens from the multi-task dataset. This serves as a baseline for comparison with the multi-task models on the ASR task. **Evaluation** It is evaluated with WER. For `TokenVerse` models, we remove task tokens from both the reference and hypothesis to compute WER for a fair comparison. We also report WER including task tokens, which reflects its prediction errors.

Named-Entity Recognition We finetune pretrained BERT[10] [87] model on our datasets for subword-level NER classification. We evaluate the models on both reference and hypothesis from the ASR model. **Evaluation** NER systems are usually evaluated by comparing their outputs against human annotations, either using an exact-match or soft-match approach [333]. We adapted these metrics to a scenario where the text comes from an ASR system. *Exact-Match:* Let $P = \{P_1, P_2, \ldots, P_n\}$ be the set of predicted entities, and $A = \{A_1, A_2, \ldots, A_n\}$ be the set of actual entities, where each P_i and A_i is accompanied by its corresponding `[NE]-[/NE]` tokens (See Figure 6.7). Thus, an entity P_i is considered correctly identified if and only if: $\forall i \in \{1, 2, \ldots, n\}, P_i = A_i$, including the tokens. *Soft-Match:* in this case we only count for the paired sets of `[NE]-[/NE]` tokens without considering if the predicted entity value P_i was correctly transcribed. After obtaining each pair, we evaluate NER with F1-score.

Speaker Change Detection For the SCD baseline, we utilize the diarization pipeline[11] from PyAnnote [348] to extract speaker change timestamps from the audio. In literature, the SCD is predominantly regarded as a task within the audio domain [346], we opt not to establish an independent text-based baseline for this task. **Evaluation** We evaluate SCD in two ways: text-based (only valid for `TokenVerse`) and time-based. In text-based evaluation, we align the reference and hypothesis using edit-distance. For each occurrence of the `[SCD]` token in the reference, matching with the same token in the hypothesis counts as True Positive; else, False Negative. Unmatched tokens in the hypothesis are considered False Positive. F1 score is calculated by standard definitions. In time-based evaluation, we obtain the timestamps where `[SCD]` tokens are predicted in the hypothesis. We calculate F1 score [338], using a collar of 250ms during timestamp matching, following common practice in speaker diarization literature [332]. Additionally, segment coverage, purity [346], and their F1 score are also reported. We use `pyannote.metrics` [349] to compute all time-based metrics.

Endpointing Considering semantic endpointing, we fine-tune BERT [87] for `[ENDP]` token classification on the multi-task training text, termed as BERT-ENDP. Results are reported on both reference text and hypothesis text obtained from `TokenVerse`. From the audio perspective, we use segmentation pipeline[12] from PyAnnote to obtain endpoint timestamps. **Evaluation** Endpointing is also evaluated in two ways: text-based and time-based. The text-based evaluation

[10] https://huggingface.co/google-bert/bert-base-uncased

[11] huggingface.co/pyannote/speaker-diarization-3.1

[12] huggingface.co/pyannote/segmentation-3.0

Table 6.8: WERs (%) for ASR on DefinedAI with `TokenVerse`. †task tokens are removed from both referene and hypothesis.

Exp	Model	w/ token	w/o token†
1)	ASR (baseline)	15.3	
2)	all-tasks	15.6	**14.7**
3-a)	single-`[SCD]`	15.2	15.1
3-b)	single-`[NE]`	15.3	14.7
3-c)	single-`[ENDP]`	**14.8**	14.7

follows the same approach as described previously for SCD. In the time-based evaluation, the F1 score computation also follows the same approach as for SCD. Additionally, we also report false alarms (FA), missed speech (MS), and detection error rate (DER), which are common metrics in endpointing literature [330].

6.2.6 Databases and Experimental Setup

Dataset To train `TokenVerse`, we require conversational audio data with corresponding transcripts, NER and segment timestamps, and speaker annotations. We could not find a large-scale public dataset satisfying all the tasks. Thus, we opt for a private dataset (*DefinedAI*[13]) which contains stereo-audio/transcript pairs for contact center conversations between agents and customers. We upsampled audio from 8 kHz to 16 kHz to align with the XLSR-53 model's requirements. Each segment includes transcripts, speaker ID and NE annotations, facilitating multi-task dataset preparation. This dataset spans health, banking and finance domains, which makes it particularly challenging due to variations in NEs. Additionally, we train and evaluate `TokenVerse` on the open-source *CallHome* English dataset (LDC97S42), which contains natural conversational stereo-audios between multiple speakers. The transcript includes named entities annotation. This dataset poses challenges due to its natural conversational nature, known to be hard for ASR modeling, and a large number of short segments without entities, differing from the DefinedAI dataset. Further details about these datasets are provided in Table 6.7.

Training TokenVerse We train `TokenVerse` on the multi-task dataset. It involves XLSR-transducer model, which is constructed from the Icefall's Transducer recipe[14] adapted with XLSR from fairseq [224] as the encoder. The fine-tuning uses Scaled Adam [186] and a learning rate scheduler that consists of a 500-step warmup phase followed by a decay phase directed by the number of steps and epochs. The model is optimized with pruned RNN-T loss [60]. The learning rate is set to $lr = 1.25e^{-3}$ and we train the model for 50 epochs. For each dataset, the best epoch is selected based on the WER on respective dev sets and results are presented on the eval sets. The task-transfer experiments are trained for an additional 10 epochs on the new task.

[13] https://www.defined.ai/

[14] https://github.com/k2-fsa/icefall/tree/master/egs/librispeech/ASR/zipformer

6.2.7 Results & Discussion

Automatic Speech Recognition For the *DefinedAI* (Table 6.8) set, WERs are reported both with and without task tokens in the reference and hypothesis for multi-task models. However, the baseline ASR model is trained without task tokens in transcripts, so there is no distinction between them. Including all tasks in `TokenVerse` (exp 2) leads to a 4% relative improvement in WER compared to the baseline ASR model (exp 1). For models trained on a single task (exp 3a-c), ASR results remain similar, except for SCD. When comparing WERs before and after token removal, we observe a relatively large gap between all-tasks and single-task models, potentially due to higher token insertion or deletion as compared to non-token words in the hypothesis. In single-task models, a larger gap is observed for `[NE]` as the model must accurately predict both tokens, introducing additional error sources. On the *CallHome* dataset (Table 6.10), the multi-task model with all tokens yields a 7.7% relative improvement. Overall, the results on both datasets indicate that the all-tasks `TokenVerse` improves ASR performance.

Table 6.9: `[SCD]` and `[ENDP]` time-based evaluation. FA: false alarm; MS: missed speech; DER: detection error rate. †F1-score computed from the Coverage-Purity perspective. ‡single-task model per task, i.e., SCD and ENDP.

		SCD		**EndPointing**			
Exp	**Model**	**F1**	**CP-F1**†	**F1**	**FA**	**MS**	**DER**
b-1/2)	PyAnnote	69.6	92.2	73.5	**1.1**	8.5	9.6
2)	all-tasks	79.7	**97.7**	**85.7**	4.7	**1.4**	6.1
3-a/c)	single‡	**87.5**	97.6	84.1	1.9	2.0	**3.9**

Named-Entity Recognition As expected, compared to evaluating BERT-NER on reference text, a significant degradation is observed when evaluated on hypothesis (Table 6.11) due to ASR errors [335]. In exact-match, on both the *DefinedAI* (Table 6.11) and *CallHome* (Table 6.10) test sets, the all-tasks `TokenVerse` outperforms the baseline BERT-NER models trained on their respective datasets and evaluated on hypothesis in F1 score. This is not the case for soft-match evaluation on the *DefinedAI* test set, where the F1 score is similar. This degradation is mostly attributed to the incorrect prediction of `[/NE]` tag by the baseline, resulting in only a partial match of the named entity words. The absolute F1 score is low on the *CallHome* dataset due to higher ASR errors on named entities, attributed to their low repetition in the training text (see Table 6.7).

Speaker Change Detection On the *DefinedAI* (Table 6.9), including all tasks in `TokenVerse` outperforms the baseline PyAnnote model in time-based evaluations. Interestingly, models trained for single-task SCD perform better than the all-tasks model in terms of F1, but show similar results for Coverage-Purity based F1. Upon closer scrutiny, we found that including `[ENDP]` delays the prediction for `[SCD]` tokens, causing the hypothesis timestamps of these tokens to fall outside the tolerance window (250ms). Increasing the tolerance window further improves the F1 for both models, with a much higher rate of increase for the all-tasks model.

Table 6.11: Text-based performances of `TokenVerse` on the `[NE]` (exact- and soft-match) and `[ENDP]`. P: precision; R: recall. †upper-bound: BERT model evaluated on text references. ‡model trained on `[ENDP]` or `[NE]` task.

Exp	Model	**[NE]-Exact**			**[NE]-Soft**			**[ENDP]**
		@P	@R	@F1	@P	@R	@F1	@F1
BERT: fine-tuned on DefinedAI								
b-1)	Eval. on Ref.†	80.0	77.0	78.5	91.6	87.9	89.7	81.6
b-2)	Eval on Hyp.	52.9	53.0	**52.9**	82.0	81.3	**81.6**	**80.5**
2)	all-tasks	65.0	51.7	**57.6**	93.0	73.2	**81.9**	**89.9**
3-b/c)	single‡	61.7	49.9	55.2	91.4	73.3	81.4	88.5

This observation is reinforced in the text-based F1 score, where the all-tasks model achieves an F1 score of 90.3% compared to 88.5% from the single-`[SCD]` model. On the *CallHome* test set (Table 6.10), the all-tasks model outperforms the PyAnnote baseline. These evaluations suggest that excluding `[SCD]` from `TokenVerse` is preferable for precise speaker change timestamps, while including all tasks improves speaker-attributed text segmentation.

Endpointing In text-based evaluation on the *DefinedAI* (Table 6.11) and *CallHome* (Table 6.10) test sets, the all-tasks `TokenVerse` outperforms the BERT-ENDP models trained on respective datasets. Additionally, on the *DefinedAI* dataset, we evaluate the BERT-ENDP model on both reference and hypothesis to understand the effect of ASR errors on `[ENDP]` token prediction. Interestingly, we do not observe a significant degradation when evaluating on the hypothesis compared to the reference. This suggests that errors introduced by ASR may not drastically affect the semantic meaning of the sentences. In time-based evaluation on the *DefinedAI* test set (Tab 6.9), the all-tasks model outperforms the baseline PyAnnote segmentation model. However, single-task ENDP is better than including all tasks in DER due to lower false alarms.

Table 6.10: F1-score and WERs for CallHome Eval set on different tasks with `TokenVerse`. †time-based F1 score. ‡baselines are computed with PyAnnote for SCD or with fine-tuned BERT on ENDP and NER (exact-match).

Exp	ASR	SCD†	ENDP	NER
	WER (↓)	F1 (↑)	F1 (↑)	F1 (↑)
baselines‡	24.6	91.7	55.9	27.4
all-tasks	**22.7**	**92.5**	**73.3**	**30.6**

Ablation results In ASR, we observed degradation for all ablation experiments, with the largest relative degradation of 2.4% in WER when `[ENDP]` was removed. Transfer learning on any of the 3 tasks does not degrade ASR performance further. The text-based evaluations of other tasks on *DefinedAI* are reported in Figure 6.8; absolute change is calculated from the all-tasks model. Removing a task adversely affects other tasks. Specifically, for SCD and endpointing,

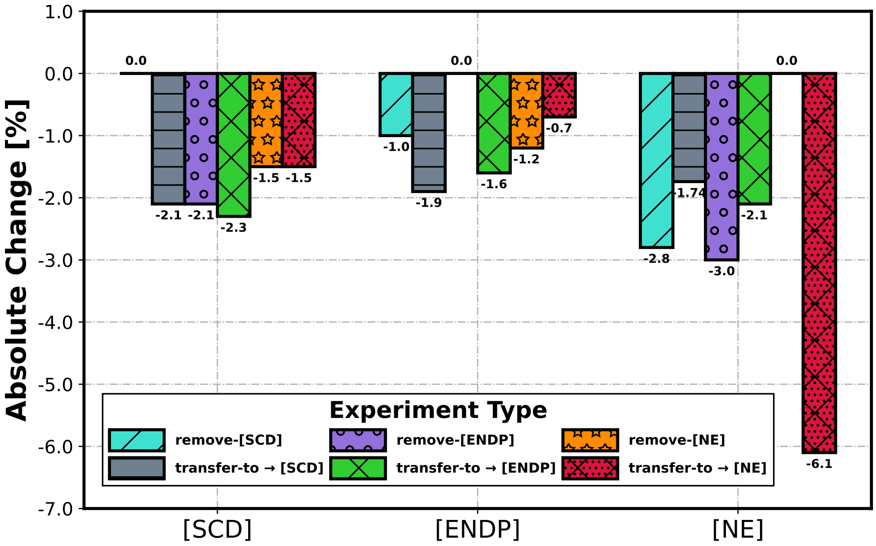


Figure 6.8: Absolute changes in text-based evaluation w.r.t all-tasks `TokenVerse` in @F1. We either remove a task, e.g., `remove-[NE]`, or transfer to the removed task, e.g., `transfer-to` → `[NE]`. Note that all-tasks `TokenVerse` performs better in all scenarios.

`[NE]` removal has the least impact on performance. Learning it afterward either improves or maintain their performance, indicating a stronger correlation between these tasks than with NER; supported by the degradation in `[SCD]` performance when `[ENDP]` is removed. Task transfer on `[ENDP]` degrades the performance further, possibly due to confusion during prediction caused by the insertion of the token before `[SCD]` during training. Transfer to NER shows relatively large degradation compared to other tasks, likely because the model must predict both `[NE]` and `[/NE]` accurately. This suggests that tasks encoded with multiple tokens may not transfer as effectively as those encoded with a single token.

Overall, all-tasks `TokenVerse` outperforms specialized models for each task and single-task models, suggesting that additional tasks improve each other. Moreover, our task transfer experiments suggest that a new task can be learned effectively.

Conclusions In this work, we demonstrate the effectiveness of a token-based multi-task model on speech and NLP using XLSR-Transducer as our ASR model, termed TokenVerse.[15] We consider speaker change detection, endpointing and named entity recognition as 3 additional tasks alongside ASR. Results on 2 datasets show that our approach improves ASR performance while outperforming strong task-specific baselines. Ablation experiments suggest that multi-task training across different domains can enhance performance on all tasks. Our approach offers flexibility for extension to numerous tasks across various domains.

[15] Further information about the XLSR-Transducer architecture in Section 4.2.

7 Conclusions and Future Directions

In the two subsections below, we conclude this thesis and outline possible future research directions connected to each domain and topic covered.

7.1 Conclusions

This thesis focuses on methodologies and techniques to develop automatic speech recognition and spoken language understanding systems under multiple low-resource settings. This includes, (1) data-, (2) compute- and (3) training-time-bounded low-resource settings. This thesis is formed as a large collection of several innovative works, where many of them are open-sourced on GitHub or shared directly with industrial partners. More specifically, this thesis focus on databases and systems such as *the ATCO2 corpus*, *BERTraffic*, *XLSR-Transducer*, *HyperConformer*, *STAC-ST*, and *TokenVerse*. Based on the outlined contributions and chapters presented in this thesis, we summarize the conclusions below:

In Chapter 2, we laid the groundwork by introducing the fundamental paradigms of ASR and SLU, highlighting hybrid-based and end-to-end ASR architectures, as well as cascaded and end-to-end SLU pipelines. This chapter covers the background needed for the rest of the thesis. It explores three domains: read and prompted speech, conversational speech, and ATC communications, with a specific emphasis on the challenges posed by the latter domain. Additionally, we discussed the evaluation metrics employed to assess ASR and SLU systems across various tasks and domains.

Chapter 3 covers the challenges of ASR applications limited by the amount of supervised data, particularly focusing on ATC communications. Through benchmarking ASR with open-source databases, we identified gaps between large-scale ASR systems and niche applications like ATC. Modern strategies leveraging pretrained foundational speech models (FSMs) were proposed to mitigate data scarcity, alongside with novel techniques that incorporate contextual information during decoding and semi-supervised training.

Chapter 4 addressed training-time and compute-bounded challenges in ASR, specifically targeting conversational speech scenarios. We introduced methods for rapid development of transducer-based streaming ASR systems, leveraging FSMs through sequence-level knowledge distillation. Effective data selection and filtering techniques were presented to enhance training efficiency and reduce computation time while achieving lower WERs. Furthermore, the XLSR-Transducer architecture open the doors to developing ASR systems for low-latency streaming settings with low supervised data. Finally, we demonstrated how the attention sink phenomena can improve WERs on challenging streaming settings.

In Chapter 5, advancements in SLU were explored across various downstream tasks, including joint speaker role and change detection and slot filing for air traffic control communications, a very challenging domain. Finally, we benchmark different representations for intent and slot-filling, including text, speech, lattice and multimodal based.

Finally, Chapter 6 concluded the thesis by examining joint ASR and SLU architectures, optimizing single models for multiple tasks such as multilingual ASR and speech-to-text translation, cross-talk detection, and acoustic-based speaker turn detection. Encoder-decoder and transducer-based models were leveraged to demonstrate multitask learning with special tokens, enabling a unified framework for ASR, speech-to-text translation, and acoustic named-entity recognition within industrial streaming applications.

7.2 Limitations and Future Directions

Test on large-scale databases Conducting experiments and evaluations on larger and more diverse speech databases beyond those currently used can provide valuable insights into the generalizability and robustness of ASR and SLU systems. Furthermore, larger datasets can help uncover performance variations across different domains, accents, and speaking styles. For instance, we noted that FSMs do not perform that well on under-resourced applications, such as ATC.

Limited number of speakers Expanding the scope of speaker variability by exploring conversations involving more than two speakers can offer a more comprehensive understanding of system performance in multi-party dialogues [101, 102]. This would require access to datasets containing conversations with multiple speakers, addressing the limitations of publicly available datasets focused on two-speaker interactions [350].

Decoder implementation of HyperConformer Further investigation into the integration and optimization of the HyperConformer architecture into the decoder of AED models could enhance the overall efficiency. So far, this architecture only supports the encoder [12]. Exploring different decoding strategies and optimizations specific to HyperConformer could lead to improved performance in real-world applications, such as low-latency streaming decoding. Another line of work, could be the integration of HyperConformer encoders into Transformer-Transducer models.

Acoustic modality for BERTraffic Exploring the utilization of acoustic modality in conjunction with *BERTraffic*, possibly through feature fusion or joint learning techniques, can enhance the representation and modeling of audio-linguistic interactions in ATC-related contexts, improving the accuracy and robustness of SLU systems. For instance, leveraging the acoustic modality can improve the speaker diarization pipeline in cases where the transcripts automatically generated with the ASR system are very noisy. Furthermore, adding contextual information, such as radar information at decoding time, can further enhance the overall model quality of *BERTraffic*.

Unified encoder-decoder and only-decoder LMs for ATC data Developing and fine-tuning unified LMs specifically tailored to ATC text. For example, (1) only-encoder, (2) encoder-decoder; or (3) only-decoder models trained with only ATC text could enhance the accuracy and contextual understanding of ASR and SLU systems in critical communication scenarios. I.e., in cases where the amount of supervised data is scarce.

Attention sink in other architectures Investigating the applicability and effectiveness of attention sink mechanisms in alternative ASR and SLU architectures beyond those explored in the current research can broaden the understanding of the attention mechanisms' impact on model performance and efficiency. For instance, the validation of attention sink on Zipformer [58], Conformer [57] or HyperConformer [12] models.

Multitasking for more tasks and languages Extending the multitasking capabilities of encoder-decoder and transformer-transducer models to support additional tasks and languages beyond those considered in the current work can lead to more versatile and adaptable speech processing systems suitable for diverse applications and environments. Examples of open source models that follow this line of research are Open Whisper models [351] and its CTC-only variant [352].

Fine-tuning of foundational speech models Continuously refining and fine-tuning FSMs with techniques like transfer learning and semi-supervised learning using domain-specific data can improve model performance and adaptability across various ASR and SLU tasks and domains. Future work should be directed to knowledge distillation techniques and better approaches to filter out hallucinated pseudo-labeled data from FSMs.

Exploration of other architectures for ASR and SLU Investigating novel architectures and frameworks for ASR and SLU, such as hybrid models combining traditional methods with deep learning approaches or leveraging graph neural networks for structured data representation, can push the boundaries of performance and scalability in speech technology applications. Emerging architectures includes (1) structured state-space models [353], (2) MAMBA [354] and (3) JAMBA [355].

Addressing these limitations and exploring these future directions can contribute to advancing the state-of-the-art in automatic speech recognition and spoken language understanding, paving the way for more robust and effective speech processing systems across different domains and use cases.

Bibliography

[1] C. Federmann and W. D. Lewis, “Microsoft speech language translation (MSLT) corpus: The IWSLT 2016 release for English, French and German,” in *Proceedings of the 13th International Conference on Spoken Language Translation*. Seattle, Washington D.C: International Workshop on Spoken Language Translation, Dec. 8-9 2016. [Online]. Available: https://aclanthology.org/2016.iwslt-1.12

[2] C. Wang, A. Wu, J. Gu, and J. Pino, “CoVoST 2 and Massively Multilingual Speech Translation,” in *Proc. Interspeech 2021*, 2021, pp. 2247–2251.

[3] J. Zuluaga-Gomez, A. Prasad, I. Nigmatulina, P. Motlicek, and M. Kleinert, “A Virtual Simulation-Pilot Agent for Training of Air Traffic Controllers,” *Aerospace*, vol. 10, no. 5, p. 490, 2023.

[4] J. Zuluaga-Gomez, I. Nigmatulina, A. Prasad, P. Motlicek, D. Khalil, S. Madikeri, A. Tart, I. Szoke, V. Lenders, M. Rigault, *et al.*, “Lessons Learned in Transcribing 5000 h of Air Traffic Control Communications for Robust Automatic Speech Understanding,” *Aerospace*, vol. 10, no. 10, p. 898, 2023.

[5] J. Zuluaga-Gomez, K. Veselý, I. Szöke, A. Blatt, P. Motlicek, M. Kocour, M. Rigault, K. Choukri, A. Prasad, S. S. Sarfjoo, *et al.*, “ATCO2 Corpus: A Large-Scale Dataset for Research on Automatic Speech Recognition and Natural Language Understanding of Air Traffic Control Communications,” *Submitted to Data-centric Machine Learning Research (DMLR) Journal, arXiv preprint arXiv:2211.04054*, 2024.

[6] J. Zuluaga-Gomez, S. Kumar, *et al.*, “Improved Streaming Transformer Transducer With Attention Sinks,” in *To be Submitted to ARR (long paper)*, 2024.

[7] I. Nigmatulina, J. Zuluaga-Gomez, *et al.*, “Fast Streaming Transducer ASR Prototyping via Knowledge Distillation with Whisper,” in *Submitted to EMNLP 2024 (long paper). **[Equal contribution]***, 2024.

[8] J. Zuluaga-Gomez, Z. Huang, X. Niu, R. Paturi, S. Srinivasan, P. Mathur, B. Thompson, and M. Federico, “End-to-End Single-Channel Speaker-Turn Aware Conversational Speech Translation,” in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 7255–7274.

[9] I. Nigmatulina, J. Zuluaga-Gomez, *et al.*, "Improved contextual adaptation with an external n-gram language model for Transducer-based ASR," in *Submitted to INTERSPEECH 2024*, 2024.

[10] S. Kumar, S. Madikeri, J. Zuluaga-Gomez, I. Nigmatulina, E. Villatoro-Tello, S. Burdisso, P. Motlicek, K. Pandia, and A. Ganapathiraju, "TokenVerse: Unifying Speech and NLP Tasks via Transducer-based ASR," in *arXiv:2407.04444*, 2024.

[11] S. Kumar, S. Madikeri, J. Zuluaga-Gomez, E. Villatoro-Tello, I. Nigmatulina, P. Motlicek, M. K. E, and A. Ganapathiraju, "XLSR-Transducer: Streaming ASR for Self-Supervised Pretrained Models," in *arXiv:2407.04439*, 2024.

[12] F. Mai, J. Zuluaga-Gomez, T. Parcollet, and P. Motlicek, "HyperConformer: Multi-head HyperMixer for Efficient Speech Recognition," in *Proc. Interspeech*, 2023, pp. 2213–2217.

[13] J. Zuluaga-Gomez, A. Prasad, I. Nigmatulina, S. S. Sarfjoo, P. Motlicek, M. Kleinert, H. Helmke, O. Ohneiser, and Q. Zhan, "How Does Pre-trained Wav2Vec 2.0 Perform on Domain-Shifted ASR? An Extensive Benchmark on Air Traffic Control Communications," in *IEEE Spoken Language Technology Workshop (SLT).* IEEE, 2023, pp. 205–212.

[14] J. Zuluaga-Gomez, S. S. Sarfjoo, A. Prasad, I. Nigmatulina, P. Motlicek, K. Ondrej, O. Ohneiser, and H. Helmke, "BERTRAFFIC: Bert-based joint speaker role and speaker change detection for air traffic control communications," in *IEEE Spoken Language Technology Workshop (SLT).* IEEE, 2023, pp. 633–640.

[15] I. Nigmatulina, J. Zuluaga-Gomez, A. Prasad, S. S. Sarfjoo, and P. Motlicek, "A two-step approach to leverage contextual data: speech recognition in air-traffic communications," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2022, pp. 6282–6286.

[16] A. Prasad, J. Zuluaga-Gomez, P. Motlicek, S. Sarfjoo, I. Nigmatulina, and K. Veselý, "Speech and Natural Language Processing Technologies for Pseudo-Pilot Simulator," in *12th SESAR Innovation Days.* Sesar Joint Undertaking., 2022.

[17] J. Zuluaga-Gomez, I. Nigmatulina, A. Prasad, P. Motlicek, K. Veselý, M. Kocour, and I. Szöke, "Contextual Semi-Supervised Learning: An Approach to Leverage Air-Surveillance and Untranscribed ATC Data in ASR Systems," in *Proc. Interspeech*, 2021, pp. 3296–3300.

[18] J. Zuluaga-Gomez, K. Veselý, A. Blatt, P. Motlicek, D. Klakow, A. Tart, I. Szöke, A. Prasad, S. Sarfjoo, P. Kolčárek, *et al.*, "Automatic call sign detection: Matching air surveillance data with air traffic spoken communications," in *Proceedings*, vol. 59, no. 1. MDPI, 2020, p. 14.

[19] J. Zuluaga-Gomez, P. Motlicek, Q. Zhan, K. Veselý, and R. Braun, "Automatic Speech Recognition Benchmark for Air-Traffic Communications," in *Proc. Interspeech*, 2020, pp. 2297–2301.

[20] J. Zuluaga-Gomez, S. Ahmed, D. Visockas, and C. Subakan, "CommonAccent: Exploring Large Acoustic Pretrained Models for Accent Classification Based on Common Voice," in *Proc. Interspeech*, 2023, pp. 5291–5295.

[21] M. Kocour, K. Veselý, A. Blatt, J. Zuluaga-Gomez, I. Szöke, J. Černocký, D. Klakow, and P. Motlicek, "Boosting of Contextual Information in ASR for Air-Traffic Call-Sign Recognition," in *Proc. Interspeech*, 2021, pp. 3301–3305.

[22] M. Kocour, K. Veselý, I. Szöke, S. Kesiraju, J. Zuluaga-Gomez, A. Blatt, A. Prasad, I. Nigmatulina, P. Motlíček, D. Klakow, *et al.*, "Automatic processing pipeline for collecting and annotating air-traffic voice communication data," *Engineering Proceedings*, vol. 13, no. 1, p. 8, 2021.

[23] M. Rigault, C. Cevenini, K. Choukri, M. Kocour, K. Veselý, I. Szoke, P. Motlicek, J. Zuluaga-Gomez, A. Blatt, D. Klakow, *et al.*, "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*, 2022, pp. 79–83.

[24] H. Helmke, K. Ondřej, S. Shetty, H. Arilíusson, T. S. Simiganosch, M. Kleinert, O. Ohneiser, H. Ehr, and J. Zuluaga-Gomez, "Readback Error Detection by Automatic Speech Recognition and Understanding-Results of HAAWAII project for Isavia's Enroute Airspace," *12th SESAR Innovation Days.*, 2022.

[25] H. Helmke, M. Kleinert, N. Ahrenhold, H. Ehr, T. Mühlhausen, O. Ohneiser, L. Klamert, P. Motlicek, A. Prasad, J. Zuluaga-Gomez, *et al.*, "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)*, 2023.

[26] I. Nigmatulina, S. Madikeri, E. Villatoro-Tello, P. Motlicek, J. Zuluaga-Gomez, K. Pandia, and A. Ganapathiraju, "Implementing Contextual Biasing in GPU Decoder for Online ASR," in *Proc. Interspeech*, 2023, pp. 4494–4498.

[27] E. Villatoro-Tello, S. Madikeri, J. Zuluaga-Gomez, B. Sharma, S. S. Sarfjoo, I. Nigmatulina, P. Motlicek, A. V. Ivanov, and A. Ganapathiraju, "Effectiveness of text, acoustic, and lattice-based representations in spoken language understanding tasks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2023, pp. 1–5.

[28] M. Ravanelli, T. Parcollet, A. Moumen, S. de Langen, C. Subakan, P. Plantinga, Y. Wang, P. Mousavi, L. D. Libera, A. Ploujnikov, F. Paissan, D. Borra, S. Zaiem, Z. Zhao, S. Zhang, G. Karakasidis, S.-L. Yeh, A. Rouhe, R. Braun, F. Mai, J. Zuluaga-Gomez, *et al.*, "Open-Source Conversational AI with SpeechBrain 1.0," in *arXiv:2407.00463*, 2024.

[29] I. Nigmatulina, R. Braun, J. Zuluaga-Gomez, and P. Motlicek, “Improving callsign recognition with air-surveillance data in air-traffic communication,” Idiap Research Institute. Idiap Research Institute, 2021, pp. 1–5.

[30] D. Khalil, A. Prasad, P. Motlicek, J. Zuluaga-Gomez, I. Nigmatulina, S. Madikeri, and C. Schuepbach, “An Automatic Speaker Clustering Pipeline for the Air Traffic Communication Domain,” *Aerospace*, vol. 10, no. 10, p. 876, 2023.

[31] N. Ahrenhold, H. Helmke, T. Mühlhausen, O. Ohneiser, M. Kleinert, H. Ehr, L. Klamert, and J. Zuluaga-Gómez, “Validating Automatic Speech Recognition and Understanding for Pre-Filling Radar Labels—Increasing Safety While Reducing Air Traffic Controllers’ Workload,” *Aerospace*, vol. 10, no. 6, p. 538, 2023.

[32] S. Burdisso, J. Zuluaga-Gomez, E. Villatoro-Tello, M. Fajcik, M. Singh, P. Smrz, and P. Motlicek, “IDIAPers@ Causal News Corpus 2022: Efficient Causal Relation Identification Through a Prompt-based Few-shot Approach,” in *Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE)*, 2022, pp. 61–69.

[33] M. Fajcik, M. Singh, J. Zuluaga-Gomez, E. Villatoro-Tello, S. Burdisso, P. Motlicek, and P. Smrz, “IDIAPers@ Causal News Corpus 2022: Extracting Cause-Effect-Signal Triplets via Pre-trained Autoregressive Language Model,” in *Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE)*, 2022, pp. 70–78.

[34] A. Prasad, J. Zuluaga-Gomez, P. Motlicek, S. Sarfjoo, I. Nigmatulina, O. Ohneiser, and H. Helmke, “Grammar Based Speaker Role Identification for Air Traffic Control Speech Recognition,” *12th SESAR Innovation Days.*, 2022.

[35] Q. Zhan, X. Xie, C. Hu, J. Zuluaga-Gomez, J. Wang, and H. Cheng, “Domain-Adversarial Based Model with Phonological Knowledge for Cross-Lingual Speech Recognition,” *Electronics*, vol. 10, no. 24, p. 3172, 2021.

[36] S. Madikeri, S. Tong, J. Zuluaga-Gomez, A. Vyas, P. Motlicek, and H. Bourlard, “Pkwrap: a pytorch package for lf-mmi training of acoustic models,” *arXiv preprint arXiv:2010.03466*, 2020.

[37] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *IEEE Transactions on audio, speech, and language processing*, vol. 20, no. 1, pp. 30–42, 2011.

[38] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, “Sequence-discriminative training of deep neural networks.” in *Proc. Interspeech 2013*, vol. 2013, 2013, pp. 2345–2349.

[39] F. Jelinek, “Continuous speech recognition by statistical methods,” *Proceedings of the IEEE*, vol. 64, no. 4, pp. 532–556, 1976.

[40] H. A. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*. Springer Science & Business Media, 1993, vol. 247.

[41] N. Morgan, H. Bourlard, S. Renals, M. Cohen, and H. Franco, "Hybrid neural network/hidden markov model systems for continuous speech recognition," in *Advances in Pattern Recognition Systems Using Neural Network Technologies*. World Scientific, 1993, pp. 255–272.

[42] S. J. Young, N. Russell, and J. Thornton, *Token passing: a simple conceptual model for connected speech recognition systems*. Cambridge University Engineering Department Cambridge, UK, 1989.

[43] P. Koen, "Pharaoh: a beam search decoder for phrase-based statistical machine translation models," in *Proceedings of the 6th Conference of the Association for Machine Translation in the Americas: Technical Papers*, 2004, pp. 115–124.

[44] D. Povey, M. Hannemann, G. Boulianne, L. Burget, A. Ghoshal, M. Janda, M. Karafiát, S. Kombrink, P. Motlíček, Y. Qian, *et al.*, "Generating exact lattices in the wfst framework," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 4213–4216.

[45] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[46] A. Graves and A. Graves, "Connectionist Temporal Classification," *Supervised sequence labelling with recurrent neural networks*, pp. 61–93, 2012.

[47] W. Chan, D. Park, C. Lee, Y. Zhang, Q. Le, and M. Norouzi, "Speechstew: Simply mix all available speech recognition data to train one large neural network," *arXiv preprint arXiv:2104.02133*, 2021.

[48] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised Pre-Training for Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 3465–3469.

[49] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[50] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino, A. Baevski, A. Conneau, and M. Auli, "XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale," in *Proc. Interspeech 2022*, 2022, pp. 2278–2282.

[51] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised Cross-Lingual Representation Learning for Speech Recognition," in *Proc. Interspeech 2021*, 2021, pp. 2426–2430.

[52] L. Lugosch, “Sequence-to-sequence learning with transducers,” Nov 2020. [Online]. Available: https://lorenlugosch.github.io/posts/2020/11/transducer/

[53] A. Graves, “Sequence transduction with recurrent neural networks,” *arXiv preprint arXiv:1211.3711*, 2012.

[54] T. N. Sainath, Y. He, B. Li, A. Narayanan, R. Pang, A. Bruguier, S.-y. Chang, W. Li, R. Alvarez, Z. Chen, *et al.*, “A streaming on-device end-to-end model surpassing server-side conventional model quality and latency,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6059–6063.

[55] M. Ghodsi, X. Liu, J. Apfel, R. Cabrera, and E. Weinstein, “RNN-transducer with stateless prediction network,” in *ICASSP*. IEEE, 2020, pp. 7049–7053.

[56] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[57] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, “Conformer: Convolution-augmented Transformer for Speech Recognition,” in *Proc. Interspeech 2020*, 2020, pp. 5036–5040.

[58] Z. Yao, L. Guo, X. Yang, W. Kang, F. Kuang, Y. Yang, Z. Jin, L. Lin, and D. Povey, “Zipformer: A faster and better encoder for automatic speech recognition,” in *The Twelfth International Conference on Learning Representations*, 2023.

[59] D. Rekesh, N. R. Koluguri, S. Kriman, S. Majumdar, V. Noroozi, H. Huang, O. Hrinchuk, K. Puvvada, A. Kumar, J. Balam, *et al.*, “Fast Conformer with linearly scalable attention for efficient speech recognition,” in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.

[60] F. Kuang, L. Guo, W. Kang, L. Lin, M. Luo, Z. Yao, and D. Povey, “Pruned RNN-T for fast, memory-efficient ASR training,” in *Proc. Interspeech 2022*, 2022, pp. 2068–2072.

[61] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[62] C.-F. Yeh, J. Mahadeokar, K. Kalgaonkar, Y. Wang, D. Le, M. Jain, K. Schubert, C. Fuegen, and M. L. Seltzer, “Transformer-transducer: End-to-end speech recognition with self-attention,” *arXiv preprint arXiv:1910.12977*, 2019.

[63] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, “Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-T loss,” in *ICASSP*. IEEE, 2020, pp. 7829–7833.

[64] B. Li, A. Gulati, J. Yu, T. N. Sainath, C.-C. Chiu, A. Narayanan, S.-Y. Chang, R. Pang, Y. He, J. Qin, *et al.*, “A better and faster end-to-end model for streaming ASR,” in *ICASSP*. IEEE, 2021, pp. 5634–5638.

[65] V. Noroozi, S. Majumdar, A. Kumar, J. Balam, and B. Ginsburg, “Stateful FastConformer with Cache-based Inference for Streaming Automatic Speech Recognition,” *arXiv preprint arXiv:2312.17279*, 2023.

[66] B. Li, S.-y. Chang, T. N. Sainath, R. Pang, Y. He, T. Strohman, and Y. Wu, “Towards fast and accurate streaming end-to-end ASR,” in *ICASSP*. IEEE, 2020, pp. 6069–6073.

[67] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. [Online]. Available: https://aclanthology.org/P16-1162

[68] T. Kudo and J. Richardson, “SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, E. Blanco and W. Lu, Eds. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 66–71. [Online]. Available: https://aclanthology.org/D18-2012

[69] D. Le, G. Keren, J. Chan, J. Mahadeokar, C. Fuegen, and M. L. Seltzer, “Deep shallow fusion for RNN-T personalization,” in *Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 251–257.

[70] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, “Hybrid CTC/attention architecture for end-to-end speech recognition,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[71] S. Kim, T. Hori, and S. Watanabe, “Joint CTC-attention based end-to-end speech recognition using multi-task learning,” in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 4835–4839.

[72] B. Yan, S. Dalmia, Y. Higuchi, G. Neubig, F. Metze, A. W. Black, and S. Watanabe, “CTC alignments improve autoregressive translation,” in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 1623–1639. [Online]. Available: https://aclanthology.org/2023.eacl-main.119

[73] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint arXiv:1807.03748*, 2018.

[74] A. Baevski, S. Schneider, and M. Auli, “vq-wav2vec: Self-supervised learning of discrete speech representations,” in *International Conference on Learning Representations*, 2019.

[75] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, “Data2vec: A general framework for self-supervised learning in speech, vision and language,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 1298–1312.

[76] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, *et al.*, “Wavlm: Large-scale self-supervised pre-training for full stack speech processing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[77] A. Baevski, W.-N. Hsu, A. Conneau, and M. Auli, “Unsupervised speech recognition,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 826–27 839, 2021.

[78] A. Bapna, C. Cherry, Y. Zhang, Y. Jia, M. Johnson, Y. Cheng, S. Khanuja, J. Riesa, and A. Conneau, “mslam: Massively multilingual joint pre-training for speech and text,” *arXiv preprint arXiv:2202.01374*, 2022.

[79] L. Barrault, Y.-A. Chung, M. C. Meglioli, D. Dale, N. Dong, P.-A. Duquenne, H. Elsahar, H. Gong, K. Heffernan, J. Hoffman, *et al.*, “Seamlessm4t-massively multilingual & multimodal machine translation,” *arXiv preprint arXiv:2308.11596*, 2023.

[80] V. Pratap, A. Tjandra, B. Shi, P. T. A. B. S. Kundu, A. Elkahky, Z. N. A. V. M. Fazel, Z. A. Baevski, and M. Auli, “Scaling speech technology to 1,000+ languages,” *arXiv preprint arXiv:2305.13516*, 2023.

[81] E. Simonnet, S. Ghannay, N. Camelin, Y. Estève, and R. De Mori, “Asr error management for improving spoken language understanding,” in *Proc. Interspeech 2017*, 2017, pp. 3329–3333.

[82] L. Lugosch, M. Ravanelli, *et al.*, “Speech Model Pre-Training for End-to-End Spoken Language Understanding,” in *Proc. Interspeech 2019*, 2019, pp. 814–818. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2396

[83] D. Serdyuk, Y. Wang, *et al.*, “Towards end-to-end spoken language understanding,” in *Proc. of ICASSP*. IEEE, 2018, pp. 5754–5758.

[84] B. Sharma, M. C. Madhavi, and H. Li, “Leveraging acoustic and linguistic embeddings from pretrained speech and language models for intent classification,” in *Proc. of ICASSP, 2021*. IEEE, 2021.

[85] P. Haghani, A. Narayanan, M. Bacchiani, *et al.*, “From audio to semantics: Approaches to end-to-end spoken language understanding,” in *IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 720–726.

[86] P. Denisov and N. T. Vu, “Pretrained semantic speech embeddings for end-to-end spoken language understanding via cross-modal teacher-student learning,” *arXiv preprint arXiv:2007.01836*, 2020.

[87] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[88] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” *arXiv preprint arXiv:1907.11692*, 2019.

[89] P. He, X. Liu, J. Gao, and W. Chen, “DeBERTa: Decoding-enhanced BERT with Disentangled Attention,” in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=XPZIaotutsD

[90] Q. Chen, Z. Zhuo, and W. Wang, “BERT for joint intent classification and slot filling,” *arXiv preprint arXiv:1902.10909*, 2019.

[91] J. Piskorski, L. Pivovarova, J. Šnajder, J. Steinberger, and R. Yangarber, “The first cross-lingual challenge on recognition, normalization, and matching of named entities in Slavic languages,” in *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing*, 2017, pp. 76–85.

[92] V. Yadav and S. Bethard, “A survey on recent advances in named entity recognition from deep learning models,” in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 2145–2158.

[93] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” *Journal of machine learning research*, vol. 12, pp. 2493–2537, 2011.

[94] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an ASR corpus based on public domain audio books,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2015, pp. 5206–5210.

[95] G. Chen, S. Chai, G.-B. Wang, J. Du, W.-Q. Zhang, C. Weng, D. Su, D. Povey, J. Trmal, J. Zhang, M. Jin, S. Khudanpur, S. Watanabe, S. Zhao, W. Zou, X. Li, X. Yao, Y. Wang, Z. You, and Z. Yan, “GigaSpeech: An Evolving, Multi-Domain ASR Corpus with 10,000 Hours of Transcribed Audio,” in *Proc. Interspeech 2021*, 2021, pp. 3670–3674.

[96] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Esteve, “TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation,” in *Speech and Computer: 20th International Conference, SPECOM 2018, Leipzig, Germany, September 18–22, 2018, Proceedings 20.* Springer, 2018, pp. 198–208.

[97] R. Ardila, M. Branson, K. Davis, M. Kohler, J. Meyer, M. Henretty, R. Morais, L. Saunders, F. Tyers, and G. Weber, “Common Voice: A Massively-Multilingual Speech Corpus,” in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 2020, pp. 4218–4222.

[98] G. Kumar, M. Post, D. Povey, and S. Khudanpur, "Some insights from translating conversational telephone speech," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 3231–3235.

[99] A. Rousseau, P. Deléglise, and Y. Esteve, "TED-LIUM: an Automatic Speech Recognition dedicated corpus," in *LREC*, 2012, pp. 125–129.

[100] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, *et al.*, "The Kaldi speech recognition toolkit," in *IEEE workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[101] L. Lu, N. Kanda, J. Li, and Y. Gong, "Streaming end-to-end multi-talker speech recognition," *IEEE Signal Processing Letters*, vol. 28, pp. 803–807, 2021.

[102] D. Raj, D. Povey, and S. Khudanpur, "SURT 2.0: Advances in Transducer-based Multi-talker Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[103] S. Watanabe, M. Mandel, J. Barker, E. Vincent, A. Arora, X. Chang, S. Khudanpur, V. Manohar, D. Povey, D. Raj, *et al.*, "Chime-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings," *arXiv preprint arXiv:2004.09249*, 2020.

[104] S. Cornell, M. Wiesner, S. Watanabe, D. Raj, X. Chang, P. Garcia, Y. Masuyama, Z.-Q. Wang, S. Squartini, and S. Khudanpur, "The chime-7 dasr challenge: Distant meeting transcription with multiple devices in diverse scenarios," *arXiv preprint arXiv:2306.13734*, 2023.

[105] N. Kanda, Y. Gaur, X. Wang, Z. Meng, and T. Yoshioka, "Serialized Output Training for End-to-End Overlapped Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 2797–2801.

[106] J. Wu, N. Kanda, T. Yoshioka, R. Zhao, Z. Chen, and J. Li, "t-SOT FNT: Streaming Multi-talker ASR with Text-only Domain Adaptation Capability," *arXiv preprint arXiv:2309.08131*, 2023.

[107] Y. Liang, F. Yu, Y. Li, P. Guo, S. Zhang, Q. Chen, and L. Xie, "BA-SOT: Boundary-Aware Serialized Output Training for Multi-Talker ASR," *Proc. Interspeech 2023*, 2023. [Online]. Available: https://arxiv.org/abs/2305.13716

[108] Y. Zou, L. Zhao, Y. Kang, J. Lin, M. Peng, Z. Jiang, C. Sun, Q. Zhang, X. Huang, and X. Liu, "Topic-oriented spoken dialogue summarization for customer service with saliency-aware topic modeling," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 16, 2021, pp. 14 665–14 673.

[109] J. Mamou, D. Carmel, and R. Hoory, "Spoken document retrieval from call-center conversations," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006, pp. 51–58.

[110] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, *et al.*, "The AMI meeting corpus," in *Proceedings of the 5th international conference on methods and techniques in behavioral research*, vol. 88. Citeseer, 2005, p. 100.

[111] Y. Lin, "Spoken Instruction Understanding in Air Traffic Control: Challenge, Technique, and Application," *Aerospace*, vol. 8, no. 3, p. 65, 2021.

[112] J. M. Cordero, M. Dorado, and J. M. de Pablo, "Automated speech recognition in ATC environment," in *Proceedings of the 2nd International Conference on Application and Theory of Automation in Command and Control Systems*, 2012, pp. 46–53.

[113] J. Ferreiros, J. Pardo, R. De Córdoba, J. Macias-Guarasa, J. Montero, F. Fernández, V. Sama, G. González, *et al.*, "A speech interface for air traffic control terminals," *Aerospace Science and Technology*, vol. 21, no. 1, pp. 7–15, 2012.

[114] M. Kleinert, H. Helmke, G. Siol, H. Ehr, A. Cerna, C. Kern, D. Klakow, P. Motlicek, Y. Oualil, M. Singh, *et al.*, "Semi-supervised adaptation of assistant based speech recognition models for different approach areas," in *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC).* IEEE, 2018, pp. 1–10.

[115] A. Srinivasamurthy, P. Motlicek, I. Himawan, G. Szaszak, Y. Oualil, and H. Helmke, "Semi-supervised learning with semantic knowledge extraction for improved speech recognition in air traffic control," in *Proc. of the 18th Annual Conference of the International Speech Communication Association*, 2017.

[116] E. Delpech, M. Laignelet, C. Pimm, C. Raynal, M. Trzos, A. Arnold, and D. Pronto, "A Real-life, French-accented Corpus of Air Traffic Control Communications," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[117] L. Graglia, B. Favennec, and A. Arnoux, "Vocalise: Assessing the impact of data link technology on the R/T channel," in *24th Digital Avionics Systems Conference*, vol. 1. IEEE, 2005, pp. 5–C.

[118] "Linguistic analysis of English phraseology and plain language in air-ground communication, author=Lopez, Stéphanie and Condamines, Anne and Josselin-Leray, Amélie and O'Donoghue, Mike and Salmon, Rupert, journal=Journal of Air Transport Studies, volume=4, number=1, pages=44–60, year=2013."

[119] K. Hofbauer, S. Petrik, and H. Hering, "The ATCOSIM Corpus of Non-Prompted Clean Air Traffic Control Speech," in *LREC*, 2008.

[120] L. Šmídl, J. Švec, D. Tihelka, J. Matoušek, J. Romportl, and P. Ircing, "Air traffic control communication (ATCC) speech corpora and their use for ASR and TTS development," *Language Resources and Evaluation*, vol. 53, no. 3, pp. 449–464, 2019.

[121] J. Godfrey, "The Air Traffic Control Corpus (ATC0) - LDC94S14A," 1994. [Online]. Available: https://catalog.ldc.upenn.edu/LDC94S14A

[122] J. Segura, T. Ehrette, A. Potamianos, D. Fohr, I. Illina, P. Breton, V. Clot, R. Gemello, M. Matassoni, and P. Maragos, "The HIWIRE database, a noisy and non-native English speech corpus for cockpit communication," *Online. http://www. hiwire. org*, 2007.

[123] B. Yang, X. Tan, Z. Chen, B. Wang, M. Ruan, D. Li, Z. Yang, X. Wu, and Y. Lin, "ATCSpeech: A Multilingual Pilot-Controller Speech Corpus from Real Air Traffic Control Environment," in *Proc. Interspeech 2020*, 2020, pp. 399–403. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1020

[124] Y. Lin, B. Yang, D. Guo, and P. Fan, "Towards multilingual end-to-end speech recognition for air traffic control," *IET Intelligent Transport Systems*, vol. 15, no. 9, pp. 1203–1214, 2021.

[125] I. Szöke, S. Kesiraju, O. Novotný, M. Kocour, K. Veselý, and J. Černocký, "Detecting English Speech in the Air Traffic Control Voice Communication," in *Proc. Interspeech 2021*, 2021, pp. 3286–3290.

[126] B. Beek, E. Neuberg, and D. Hodge, "An assessment of the technology of automatic speech recognition for military applications," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 4, pp. 310–322, 1977.

[127] C. J. Hamel, D. Kotick, and M. Layton, "Microcomputer system integration for air control training," Naval Training Systems Center, Orlando FL, Tech. Rep., 1989.

[128] K. Matrouf, J. Gauvain, F. Neel, and J. Mariani, "Adapting probability-transitions in DP matching processing for an oral task-oriented dialogue," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP).* IEEE, 1990, pp. 569–572.

[129] R. Tarakan, K. Baldwin, and N. Rozen, "An automated simulation pilot capability to support advanced air traffic controller training," in *The 26th Congress of ICAS and 8th AIAA ATIO*, 2008.

[130] J. Zhang, P. Zhang, D. Guo, Y. Zhou, Y. Wu, B. Yang, and Y. Lin, "Automatic repetition instruction generation for air traffic control training using multi-task learning with an improved copy network," *Knowledge-Based Systems*, vol. 241, p. 108232, 2022.

[131] Y. Lin, Y. Wu, D. Guo, P. Zhang, C. Yin, B. Yang, and J. Zhang, "A deep learning framework of autonomous pilot agent for air traffic controller training," *IEEE transactions on human-machine systems*, vol. 51, no. 5, pp. 442–450, 2021.

[132] H. Helmke, O. Ohneiser, T. Mühlhausen, and M. Wies, "Reducing controller workload with automatic speech recognition," in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC).* IEEE, 2016, pp. 1–10.

[133] H. Helmke, O. Ohneiser, J. Buxbaum, and C. Kern, “Increasing ATM efficiency with assistant based speech recognition,” in *Proc. of the 13th USA/Europe Air Traffic Management Research and Development Seminar, Seattle, USA*, 2017.

[134] M. Kleinert, H. Helmke, S. Shetty, O. Ohneiser, H. Ehr, A. Prasad, P. Motlicek, and J. Harfmann, “Automated Interpretation of Air Traffic Control Communication: The Journey from Spoken Words to a Deeper Understanding of the Meaning,” in *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*. IEEE, 2021, pp. 1–9.

[135] O. Ohneiser, H. Helmke, S. Shetty, M. Kleinert, H. Ehr, Š. Murauskas, T. Pagirys, G. Balogh, A. Tønnesen, G. Kis-Pál, *et al.*, “Understanding tower controller communication for support in Air Traffic Control displays,” *Proceedings of the 12th SESAR Innovation Days, Budapest, Hungary*, pp. 5–8, 2022.

[136] N. Ryant *et al.*, “The Second DIHARD Diarization Challenge: Dataset, Task, and Baselines,” in *Proc. Interspeech 2019*, 2019, pp. 978–982.

[137] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[138] D. Imseng, B. Potard, P. Motlicek, A. Nanchen, and H. Bourlard, “Exploiting untranscribed foreign data for speech recognition in well-resourced languages,” in *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2014, pp. 2322 – 2326.

[139] B. Khonglah, S. Madikeri, S. Dey, H. Bourlard, P. Motlicek, and J. Billa, “Incremental Semi-Supervised Learning For Multi-Genre Speech Recognition,” in *Proceedings of ICASSP 2020*, 2020.

[140] LiveATC, “LiveATC.net - Live Air Traffic,” 2020. [Online]. Available: https://www.liveatc.net/

[141] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, “Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks,” in *Proc. Interspeech 2018*, 2018, pp. 3743–3747.

[142] T. Pellegrini, J. Farinas, E. Delpech, and F. Lancelot, “The Airbus Air Traffic Control speech recognition 2018 challenge: towards ATC automatic transcription and call sign detection,” *arXiv preprint arXiv:1810.12614*, 2018.

[143] J. R. Novak, N. Minematsu, and K. Hirose, “Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-gram models in the WFST framework,” *Nat. Lang. Eng.*, vol. 22, no. 6, pp. 907–938, 2016.

[144] J. Drexler and J. Glass, “Subword regularization and beam search decoding for end-to-end automatic speech recognition,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6266–6270.

[145] A. Zeyer, K. Irie, R. Schlüter, and H. Ney, "Improved training of end-to-end attention models for speech recognition," *CoRR*, vol. abs/1805.03294, 2018. [Online]. Available: http://arxiv.org/abs/1805.03294

[146] Z. Zeng, Y. Khassanov, V. T. Pham, H. Xu, E. S. Chng, and H. Li, "On the End-to-End Solution to Mandarin-English Code-Switching Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2165–2169.

[147] A. Stolcke, "Srilm-an extensible language modeling toolkit," in *Seventh international conference on spoken language processing*, 2002.

[148] Z.-Q. Zhang, Y. Song, M.-H. Wu, X. Fang, and L.-R. Dai, "XLST: Cross-lingual Self-training to Learn Multilingual Representation for Low Resource Speech Recognition," *arXiv preprint arXiv:2103.08207*, 2021.

[149] W.-N. Hsu, A. Sriram, A. Baevski, T. Likhomanenko, Q. Xu, V. Pratap, J. Kahn, A. Lee, R. Collobert, G. Synnaeve, *et al.*, "Robust wav2vec 2.0: Analyzing Domain Shift in Self-Supervised Pre-Training," *arXiv preprint arXiv:2104.01027*, 2021.

[150] Y. Meng, Y.-H. Chou, A. T. Liu, and H.-y. Lee, "Don't speak too fast: The impact of data bias on self-supervised speech models," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 3258–3262.

[151] M. Zanon Boito, L. Besacier, N. Tomashenko, and Y. Estève, "A Study of Gender Impact in Self-supervised Models for Speech-to-Text Systems," in *Proc. Interspeech 2022*, 2022, pp. 1278–1282.

[152] M. Riviere, J. Copet, and G. Synnaeve, "ASR4REAL: An extended benchmark for speech models," *arXiv preprint arXiv:2110.08583*, 2021.

[153] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *Proc. Interspeech 2016*, 2016, pp. 2751–2755.

[154] J. Kahn, M. Riviere, W. Zheng, E. Kharitonov, Q. Xu, P.-E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, *et al.*, "Libri-light: A benchmark for ASR with limited or no supervision," in *ICASSP*. IEEE, 2020, pp. 7669–7673.

[155] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, and M. F. et al, "Transformers: State-of-the-art natural language processing," in *EMNLP (Demos)*, 2020, pp. 38–45.

[156] Q. Lhoest, A. V. del Moral, Y. Jernite, A. Thakur, P. von Platen, S. Patil, J. Chaumond, M. Drame, J. Plu, L. Tunstall, *et al.*, "Datasets: A community library for natural language processing," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2021, pp. 175–184.

[157] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[158] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," *arXiv preprint arXiv:1606.08415*, 2016.

[159] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7

[160] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," pp. 2613–2617, 2019. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2680

[161] N. Moritz, T. Hori, and J. Le Roux, "Streaming end-to-end speech recognition with joint CTC-attention based models," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 936–943.

[162] A. Narayanan, T. N. Sainath, R. Pang, J. Yu, C.-C. Chiu, R. Prabhavalkar, E. Variani, and T. Strohman, "Cascaded encoders for unifying streaming and non-streaming ASR," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5629–5633.

[163] L. Lu, J. Li, and Y. Gong, "Endpoint Detection for Streaming End-to-End Multi-Talker ASR," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7312–7316.

[164] A. Schmidt, Y. Oualil, O. Ohneiser, M. Kleinert, M. Schulder, A. Khan, H. Helmke, and D. Klakow, "Context-based recognition network adaptation for improving on-line ASR in air traffic control," in *2014 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2014, pp. 13–18.

[165] T. Shore, F. Faubel, H. Helmke, and D. Klakow, "Knowledge-based word lattice rescoring in a dynamic context," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[166] Y. Oualil, M. Schulder, H. Helmke, A. Schmidt, and D. Klakow, "Real-time integration of dynamic context information for improving automatic speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[167] Y. Oualil, D. Klakow, G. Szaszák, A. Srinivasamurthy, H. Helmke, and P. Motlicek, "A context-aware speech recognition and understanding system for air traffic control domain," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 404–408.

[168] K. Hall, E. Cho, C. Allauzen, F. Beaufays, N. Coccaro, K. Nakajima, M. Riley, B. Roark, D. Rybach, and L. Zhang, "Composition-based on-the-fly rescoring for salient n-gram biasing," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[169] P. Aleksic, M. Ghodsi, A. Michaely, C. Allauzen, K. Hall, B. Roark, D. Rybach, and P. Moreno, "Bringing contextual information to google speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[170] J. Serrino, L. Velikovich, P. S. Aleksic, and C. Allauzen, "Contextual recovery of out-of-lattice named entities in automatic speech recognition." in *Interspeech*, 2019, pp. 3830–3834.

[171] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech & Language*, vol. 16, no. 1, pp. 69–88, 2002.

[172] E. McDermott, H. Sak, and E. Variani, "A density ratio approach to language model fusion in end-to-end automatic speech recognition," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 434–441.

[173] N. Kanda, X. Lu, and H. Kawai, "Maximum a posteriori Based Decoding for CTC Acoustic Models," in *Proc. Interspeech 2016*, 2016, pp. 1868–1872.

[174] E. Variani, D. Rybach, C. Allauzen, and M. Riley, "Hybrid autoregressive transducer (hat)," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6139–6143.

[175] D. Zhao, T. N. Sainath, D. Rybach, P. Rondon, D. Bhatia, B. Li, and R. Pang, "Shallow-Fusion End-to-End Contextual Biasing," *Proc. Interspeech 2019*, pp. 1418–1422, 2019.

[176] A. Kannan, Y. Wu, P. Nguyen, T. N. Sainath, Z. Chen, and R. Prabhavalkar, "An analysis of incorporating an external language model into a sequence-to-sequence model," in *ICASSP*. IEEE, 2018, pp. 1–5828.

[177] N. Jung, G. Kim, and J. S. Chung, "Spell my name: keyword boosted speech recognition," in *ICASSP*. IEEE, 2022, pp. 6642–6646.

[178] Y. Guo, Z. Qiu, H. Huang, and C. E. Siong, "Improved Keyword Recognition Based on Aho-Corasick Automaton," in *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2023, pp. 1–7.

[179] M. Jain, G. Keren, J. Mahadeokar, G. Zweig, F. Metze, and Y. Saraf, "Contextual RNN-T for open domain ASR," in *Proc. Interspeech 2020*, 2020.

[180] D. Qiu, T. Munkhdalai, Y. He, and K. C. Sim, "Context-Aware Neural Confidence Estimation for Rare Word Speech Recognition," in *Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 31–37.

[181] A. V. Aho and M. J. Corasick, “Efficient string matching: an aid to bibliographic search,” *Communications of the ACM*, vol. 18, no. 6, pp. 333–340, 1975.

[182] A. Svete, B. Dayan, R. Cotterell, T. Vieira, and J. Eisner, “Algorithms for Acyclic Weighted Finite-State Automata with Failure Arcs,” in *EMNLP*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 8289–8305.

[183] B. Meyer, “Incremental string matching,” *Information Processing Letters*, vol. 21, no. 5, pp. 219–227, 1985.

[184] J. Drexler Fox and N. Delworth, “Improving Contextual Recognition of Rare Words with an Alternate Spelling Prediction Model,” in *Proc. Interspeech 2022*, 2022, pp. 3914–3918.

[185] M. D. Rio, N. Delworth, R. Westerman, M. Huang, N. Bhandari, J. Palakapilly, Q. McNamara, J. Dong, P. Zelasko, and M. Jette, “Earnings-21: A Practical Benchmark for ASR in the Wild,” in *Proc. Interspeech 2021*, 2021.

[186] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[187] M. Del Rio, P. Ha, Q. McNamara, C. Miller, and S. Chandra, “Earnings-22: A Practical Benchmark for Accents in the Wild,” *arXiv preprint arXiv:2203.15591*, 2022.

[188] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” in *International Conference on Machine Learning*. PMLR, 2023, pp. 28 492–28 518.

[189] D. Imseng, J. Dines, P. Motlicek, P. N. Garner, and H. Bourlard, “Comparing different acoustic modeling techniques for multilingual boosting,” in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[190] S. Dey, P. Motlicek, T. Bui, and F. Dernoncourt, “Exploiting semi-supervised training through a dropout regularization in end-to-end speech recognition,” *Proc. Interspeech 2019*, pp. 734–738, 2019.

[191] R. A. Braun, S. Madikeri, and P. Motlicek, “A comparison of methods for oov-word recognition on a new public dataset,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5979–5983.

[192] L. Velikovich, I. Williams, J. Scheiner, P. S. Aleksic, P. J. Moreno, and M. Riley, “Semantic lattice processing in contextual automatic speech recognition for google assistant.” in *Proc. Interspeech 2018*, 2018, pp. 2222–2226.

[193] J. Scheiner, I. Williams, and P. Aleksic, “Voice search language model adaptation using contextual information,” in *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 253–257.

[194] L. Vasserman, B. Haynor, and P. Aleksic, "Contextual language model adaptation using dynamic classes," in *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 441–446.

[195] Y. Zhang, J. Qin, D. S. Park, W. Han, C.-C. Chiu, R. Pang, Q. V. Le, and Y. Wu, "Pushing the Limits of Semi-Supervised Learning for Automatic Speech Recognition," *arXiv preprint arXiv:2010.10504*, 2020.

[196] H. Helmke, M. Slotty, M. Poiger, D. F. Herrer, O. Ohneiser, N. Vink, A. Cerna, P. Hartikainen, B. Josefsson, D. Langr, *et al.*, "Ontology for transcription of ATC speech commands of SESAR 2020 solution PJ. 16-04," in *IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*. IEEE, 2018, pp. 1–10.

[197] M. Bartelds, N. San, B. McDonnell, D. Jurafsky, and M. Wieling, "Making more of little data: Improving low-resource automatic speech recognition using data augmentation," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 715–729. [Online]. Available: https://aclanthology.org/2023.acl-long.42

[198] H. Zhu, D. Gao, G. Cheng, D. Povey, P. Zhang, and Y. Yan, "Alternative pseudo-labeling for semi-supervised automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[199] L. Lugosch, T. Likhomanenko, G. Synnaeve, and R. Collobert, "Pseudo-labeling for massively multilingual speech recognition," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7687–7691.

[200] R. Prabhavalkar, T. Hori, T. N. Sainath, R. Schlüter, and S. Watanabe, "End-to-end speech recognition: A survey," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[201] C.-C. Chiu, J. Qin, Y. Zhang, J. Yu, and Y. Wu, "Self-supervised learning with random-projection quantizer for speech recognition," in *International Conference on Machine Learning*. PMLR, 2022, pp. 3915–3924.

[202] H. Yang, Y. Shangguan, D. Wang, M. Li, *et al.*, "Omni-sparsity DNN: Fast sparsity optimization for on-device streaming E2E ASR via supernet," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8197–8201.

[203] Y. Higuchi, N. Moritz, J. L. Roux, and T. Hori, "Momentum Pseudo-Labeling for Semi-Supervised Speech Recognition," in *Proc. Interspeech 2021*, 2021, pp. 726–730.

[204] G. Zavaliagkos and T. Colthurst, "Utilizing untranscribed training data to improve perfomance." in *LREC*. Citeseer, 1998, pp. 317–322.

[205] W.-N. Hsu, Y.-H. H. Tsai, B. Bolte, R. Salakhutdinov, and A. Mohamed, "HuBERT: How much can a bad teacher benefit ASR pre-training?" in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2021, pp. 6533–6537.

[206] D. Gao, M. Wiesner, H. Xu, L. P. Garcia, D. Povey, and S. Khudanpur, "Bypass temporal classification: Weakly supervised automatic speech recognition with imperfect transcripts," *arXiv preprint arXiv:2306.01031*, 2023.

[207] T. Likhomanenko, R. Collobert, N. Jaitly, and S. Bengio, "Continuous Soft Pseudo-Labeling in ASR," in *I Can't Believe It's Not Better Workshop: Understanding Deep Learning Through Empirical Falsification*, 2022.

[208] D. Berrebbi, R. Collobert, S. Bengio, N. Jaitly, and T. Likhomanenko, "Continuous pseudo-labeling from the start," in *The Eleventh International Conference on Learning Representations*, 2022.

[209] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[210] R. Takashima, S. Li, and H. Kawai, "An investigation of a knowledge distillation method for CTC acoustic models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2018, pp. 5809–5813.

[211] Y. Chebotar and A. Waters, "Distilling Knowledge from Ensembles of Neural Networks for Speech Recognition," in *Proc. Interspeech 2016*, 2016, pp. 3439–3443.

[212] S. Watanabe, T. Hori, J. Le Roux, and J. R. Hershey, "Student-teacher network learning with enhanced features," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2017, pp. 5275–5279.

[213] S. Gandhi, P. von Platen, and A. M. Rush, "Distil-whisper: Robust knowledge distillation via large-scale pseudo labelling," *arXiv preprint arXiv:2311.00430*, 2023.

[214] T. P. Ferraz, M. Z. Boito, C. Brun, and V. Nikoulina, "Multilingual distilwhisper: Efficient distillation of multi-task speech models via language-specific experts," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.

[215] S. Panchapagesan, D. S. Park, C.-C. Chiu, Y. Shangguan, Q. Liang, and A. Gruenstein, "Efficient knowledge distillation for RNN-transducer models," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2021, pp. 5639–5643.

[216] G. Kurata and G. Saon, "Knowledge distillation from offline to streaming rnn transducer for end-to-end speech recognition." in *Proc. Interspeech 2020*, 2020, pp. 2117–2121.

[217] X. Yang, Q. Li, and P. C. Woodland, “Knowledge distillation for neural transducers from large self-supervised pre-trained models,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8527–8531.

[218] M. Bain, J. Huh, T. Han, and A. Zisserman, “WhisperX: Time-Accurate Speech Transcription of Long-Form Audio,” in *Proc. Interspeech 2023*, 2023, pp. 4489–4493.

[219] P. Żelasko, D. Povey, J. Trmal, S. Khudanpur, *et al.*, “Lhotse: a speech data representation library for the modern deep learning ecosystem,” *arXiv preprint arXiv:2110.12561*, 2021.

[220] P. Swietojanski, S. Braun, *et al.*, “Variable attention masking for configurable transformer transducer speech recognition,” in *ICASSP*. IEEE, 2023, pp. 1–5.

[221] G. Xiao, Y. Tian, B. Chen, S. Han, and M. Lewis, “Efficient streaming language models with attention sinks,” *arXiv preprint arXiv:2309.17453*, 2023.

[222] A. Vyas, S. Madikeri, and H. Bourlard, “Comparing CTC and LFMMI for out-of-domain adaptation of wav2vec 2.0 acoustic model,” in *Proceedings of Interspeech*, 2021. [Online]. Available: https://arxiv.org/abs/2104.02558

[223] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, *et al.*, “Big bird: Transformers for longer sequences,” *Advances in neural information processing systems*, vol. 33, pp. 17 283–17 297, 2020.

[224] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, “fairseq: A Fast, Extensible Toolkit for Sequence Modeling,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, 2019, pp. 48–53.

[225] G. Attanasio, B. Savoldi, D. Fucci, and D. Hovy, “Multilingual speech models for automatic speech recognition exhibit gender performance gaps,” *arXiv preprint arXiv:2402.17954*, 2024.

[226] A. Koenecke, A. S. G. Choi, K. Mei, H. Schellmann, and M. Sloane, “Careless whisper: Speech-to-text hallucination harms,” *arXiv preprint arXiv:2402.08021*, 2024.

[227] A. Mittal, R. Murthy, V. Kumar, and R. Bhat, “Towards understanding and mitigating the hallucinations in nlp and speech,” in *Proceedings of the 7th Joint International Conference on Data Science & Management of Data (11th ACM IKDD CODS and 29th COMAD)*, 2024, pp. 489–492.

[228] J. Yu, C.-C. Chiu, B. Li, S.-y. Chang, T. N. Sainath, Y. He, A. Narayanan, W. Han, A. Gulati, Y. Wu, *et al.*, “Fastemit: Low-latency streaming ASR with sequence-level emission regularization,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6004–6008.

[229] J. Kim, H. Lu, A. Tripathi, Q. Zhang, and H. Sak, “Reducing Streaming ASR Model Delay with Self Alignment,” in *Proc. Interspeech 2021*, 2021, pp. 3440–3444.

[230] M. Yang, A. Tjandra, C. Liu, D. Zhang, D. Le, and O. Kalinli, “Learning ASR pathways: A sparse multilingual ASR model,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[231] C.-I. J. Lai, Y. Zhang, A. H. Liu, S. Chang, Y.-L. Liao, Y.-S. Chuang, K. Qian, S. Khurana, D. Cox, and J. Glass, “Parp: Prune, adjust and re-prune for self-supervised speech recognition,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 21 256–21 272, 2021.

[232] A. Vyas, W.-N. Hsu, M. Auli, and A. Baevski, “On-demand compute reduction with stochastic wav2vec 2.0,” in *Proc. Interspeech 2022*, 2022, pp. 3048–3052.

[233] F. Mai, A. Pannatier, F. Fehr, H. Chen, F. Marelli, F. Fleuret, and J. Henderson, “Hypermixer: An mlp-based low cost alternative to transformers,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023.

[234] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, “Speech recognition using deep neural networks: A systematic review,” *IEEE access*, vol. 7, pp. 19 143–19 165, 2019.

[235] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang, *et al.*, “A comparative study on transformer vs rnn in speech applications,” in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 449–456.

[236] Y. Peng, S. Dalmia, I. Lane, and S. Watanabe, “Branchformer: Parallel MLP-attention architectures to capture local and global context for speech recognition and understanding,” in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 17 627–17 643. [Online]. Available: https://proceedings.mlr.press/v162/peng22a.html

[237] K. Kim, F. Wu, Y. Peng, J. Pan, P. Sridhar, K. J. Han, and S. Watanabe, “E-Branchformer: Branchformer with Enhanced merging for speech recognition,” in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 84–91.

[238] Y. Gao, J. Fernandez-Marques, T. Parcollet, A. Mehrotra, and N. Lane, “Federated Self-supervised Speech Representations: Are We There Yet?” in *Proc. Interspeech 2022*, 2022, pp. 3809–3813.

[239] T. Parcollet and M. Ravanelli, “The Energy and Carbon Footprint of Training End-to-End Speech Recognizers,” in *Proc. Interspeech 2021*, 2021, pp. 4583–4587.

[240] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, “Efficient transformers: A survey,” *ACM Computing Surveys*, vol. 55, no. 6, pp. 1–28, 2022.

[241] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, “Linformer: Self-attention with linear complexity,” *arXiv preprint arXiv:2006.04768*, 2020.

[242] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, “Transformers are rnns: Fast autoregressive transformers with linear attention,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 5156–5165.

[243] A. Vyas, A. Katharopoulos, and F. Fleuret, “Fast transformers with clustered attention,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 665–21 674, 2020.

[244] R. Child, S. Gray, A. Radford, and I. Sutskever, “Generating long sequences with sparse transformers,” *arXiv preprint arXiv:1904.10509*, 2019.

[245] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, *et al.*, “MLP-Mixer: An all-MLP Architecture for Vision,” *Advances in neural information processing systems*, vol. 34, pp. 24 261–24 272, 2021.

[246] H. Liu, Z. Dai, D. So, and Q. V. Le, “Pay attention to mlps,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 9204–9215, 2021.

[247] S. Chen, E. Xie, C. Ge, D. Liang, and P. Luo, “Cyclemlp: A mlp-like architecture for dense prediction,” *arXiv preprint arXiv:2107.10224*, 2021.

[248] “Dynamixer: a vision MLP architecture with dynamic mixing, author=Wang, Ziyu and Jiang, Wenhao and Zhu, Yiming M and Yuan, Li and Song, Yibing and Liu, Wei,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 22 691–22 701.

[249] C. Xing, D. Wang, L. Dai, Q. Liu, and A. Avila, “Speech-MLP: a simple MLP architecture for speech processing,” 2022. [Online]. Available: https://openreview.net/forum?id=-u8EliRNW8k

[250] J. Sakuma, T. Komatsu, and R. Scheibler, “Mlp-asr: Sequence-length agnostic all-mlp architectures for speech recognition,” *arXiv preprint arXiv:2202.08456*, 2022.

[251] D. Ha, A. M. Dai, and Q. V. Le, “Hypernetworks,” in *International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=rkpACe1lx

[252] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.

[253] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, “Transformer-XL: Attentive language models beyond a fixed-length context,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 2978–2988. [Online]. Available: https://aclanthology.org/P19-1285

[254] H. Helmke, M. Kleinert, S. Shetty, O. Ohneiser, H. Ehr, H. Arilíusson, T. S. Simiganoschi, A. Prasad, P. Motlicek, K. Veselý, *et al.*, "Readback Error Detection by Automatic Speech Recognition to Increase ATM Safety," in *Proceedings of the Fourteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2021), Virtual Event*, 2021, pp. 20–23.

[255] S. Vajjala and R. Balasubramaniam, "What do we Really Know about State of the Art NER?" *arXiv preprint arXiv:2205.00034*, 2022.

[256] R. Grishman and B. M. Sundheim, "Message understanding conference-6: A brief history," in *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996.

[257] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[258] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*. PMLR, 2013, pp. 1310–1318.

[259] A. Blatt, M. Kocour, K. Veselý, I. Szőke, and D. Klakow, "Call-sign recognition and understanding for noisy air-traffic transcripts using surveillance information," in *ICASSP*, 2022, pp. 8357–8361.

[260] Z. He, Z. Wang, W. Wei, S. Feng, X. Mao, and S. Jiang, "A Survey on Recent Advances in Sequence Labeling from Deep Learning Models," *arXiv preprint arXiv:2011.06727*, 2020.

[261] C. Zhou, B. Cule, and B. Goethals, "Pattern based sequence classification," *IEEE Transactions on knowledge and Data Engineering*, vol. 28, no. 5, pp. 1285–1298, 2015.

[262] M. Birjali, M. Kasri, and A. Beni-Hssane, "A comprehensive survey on sentiment analysis: Approaches, challenges and trends," *Knowledge-Based Systems*, vol. 226, p. 107134, 2021.

[263] ICAO, "ICAO phraseology reference guide," vol. 1, 2020. [Online]. Available: https://www.skybrary.aero/bookshelf/books/115.pdf

[264] F. Landini, J. Profant, M. Diez, and L. Burget, "Bayesian HMM clustering of x-vector sequences (vbx) in speaker diarization: theory, implementation and analysis on standard tasks," *Computer Speech & Language*, vol. 71, p. 101254, 2022.

[265] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with permutation-free objectives," *arXiv preprint arXiv:1909.05952*, 2019.

[266] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with self-attention," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 296–303.

[267] S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, and K. Nagamatsu, “End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors,” *arXiv preprint arXiv:2005.09921*, 2020.

[268] H. H. Mao, S. Li, J. McAuley, and G. W. Cottrell, “Speech recognition and multi-speaker diarization of long conversations,” 2020.

[269] K. Ma, C. Xiao, and J. D. Choi, “Text-based speaker identification on multiparty dialogues using multi-document convolutional neural networks,” in *Proceedings of ACL 2017, Student Research Workshop*. Association for Computational Linguistics, 2017, pp. 49–55.

[270] E. Han, C. Lee, and A. Stolcke, “Bw-eda-eend: Streaming end-to-end neural speaker diarization for a variable number of speakers,” in *ICASSP*. IEEE, 2021, pp. 7193–7197.

[271] A. Khare, E. Han, Y. Yang, and A. Stolcke, “Asr-aware end-to-end neural diarization,” in *ICASSP*. IEEE, 2022.

[272] O. Ohneiser, S. Sarfjoo, H. Helmke, S. Shetty, P. Motlicek, M. Kleinert, H. Ehr, and Š. Murauskas, “Robust Command Recognition for Lithuanian Air Traffic Control Tower Utterances,” in *Interspeech*, 2021, pp. 3291–3295.

[273] S. S. Sarfjoo, S. Madikeri, and P. Motlicek, “Speech activity detection based on multilingual speech recognition system,” in *Proc. Interspeech 2021*, 2021, p. 4369–4373.

[274] R. Jonker and T. Volgenant, “Improving the hungarian assignment algorithm,” *Operations Research Letters*, vol. 5, no. 4, pp. 171–175, 1986.

[275] E. Villatoro-Tello, S. Madikeri, P. Motlicek, A. Ganapathiraju, and A. V. Ivanov, “Expanded lattice embeddings for spoken document retrieval on informal meetings,” in *Proceedings of the 45th ACM SIGIR Conference*, 2022, p. 2669–2674. [Online]. Available: https://doi.org/10.1145/3477495.3531921

[276] G. Tür, A. Deoras, and D. Hakkani-Tür, “Semantic parsing using word confusion networks with conditional random fields,” in *Proc. Interspeech 2013*, 2013, pp. 2579–2583.

[277] C. Liu, S. Zhu, *et al.*, “Jointly Encoding Word Confusion Network and Dialogue Context with BERT for Spoken Language Understanding,” in *Proc. Interspeech 2020*, 2020, pp. 871–875. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1632

[278] J. P. McKenna, S. Choudhary, M. Saxon, G. P. Strimel, and A. Mouchtaris, “Semantic complexity in end-to-end spoken language understanding,” in *Proc. Interspeech 2020*, 2020, pp. 4273–4277.

[279] C. T. Hemphill, J. J. Godfrey, and G. R. Doddington, “The ATIS spoken language systems pilot corpus,” in *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*, 1990.

[280] A. Coucke, A. Saade, *et al.*, "Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces," *arXiv preprint arXiv:1805.10190*, 2018.

[281] E. Bastianelli, A. Vanzo, *et al.*, "SLURP: A Spoken Language Understanding Resource Package," in *Proceedings of the 2020 EMNLP*, 2020, pp. 7252–7262.

[282] A. Vanzo, E. Bastianelli, and O. Lemon, "Hierarchical multi-task natural language understanding for cross-domain conversational AI: HERMIT NLU," in *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, Sept. 2019, pp. 254–263.

[283] Y. Zou, H. Sun, and Z. Chen, "Associated lattice-bert for spoken language understanding," in *International Conference on Neural Information Processing*, 2021, pp. 579–586.

[284] M. Henderson, B. Thomson, and J. D. Williams, "The second dialog state tracking challenge," in *Proceedings SIGDIAL*, 2014, pp. 263–272.

[285] Y.-H. H. Tsai, S. Bai, *et al.*, "Multimodal transformer for unaligned multimodal language sequences," in *Proceedings of the ACL*, vol. 2019, 2019, p. 6558.

[286] Y. Shao, Y. Wang, D. Povey, and S. Khudanpur, "Pychain: A fully parallelized pytorch implementation of LF-MMI for end-to-end ASR," *arXiv preprint arXiv:2005.09824*, 2020.

[287] H. Hadian, H. Sameti, D. Povey, and S. Khudanpur, "End-to-end Speech Recognition Using Lattice-free MMI," in *Proc. Interspeech 2018*, 2018, pp. 12–16.

[288] Y. Wang *et al.*, "Espresso: A Fast End-to-end Neural Speech Recognition Toolkit," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019.

[289] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, *et al.*, "SpeechBrain: A general-purpose speech toolkit," *arXiv preprint arXiv:2106.04624*, 2021.

[290] A. Anastasopoulos, O. Bojar, *et al.*, "FINDINGS OF THE IWSLT 2021 EVALUATION CAMPAIGN," in *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*. Bangkok, Thailand (online): Association for Computational Linguistics, Aug. 2021, pp. 1–29. [Online]. Available: https://aclanthology.org/2021.iwslt-1.1

[291] A. Anastasopoulos, L. Barrault, *et al.*, "Findings of the IWSLT 2022 evaluation campaign," in *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*. Dublin, Ireland (in-person and online): Association for Computational Linguistics, May 2022, pp. 98–157. [Online]. Available: https://aclanthology.org/2022.iwslt-1.10

[292] A. Waibel, A. N. Jain, A. E. McNair, H. Saito, A. G. Hauptmann, and J. Tebelskis, "JANUS: a speech-to-speech translation system using connectionist and symbolic processing

strategies," in *1991 International Conference on Acoustics, Speech, and Signal Processing, ICASSP '91, Toronto, Ontario, Canada, May 14-17, 1991.* IEEE Computer Society, 1991, pp. 793–796. [Online]. Available: https://doi.org/10.1109/ICASSP.1991.150456

[293] E. Vidal, "Finite-state speech-to-speech translation," in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '97, Munich, Germany, April 21-24, 1997.* IEEE Computer Society, 1997, pp. 111–114. [Online]. Available: https://doi.org/10.1109/ICASSP.1997.599563

[294] F. Casacuberta, M. Federico, H. Ney, and E. Vidal, "Recent efforts in spoken language translation," *IEEE Signal Processing Magazine*, vol. 25, no. 3, pp. 80–88, 2008.

[295] A. Berard, O. Pietquin, C. Servan, and L. Besacier, "Listen and translate: A proof of concept for end-to-end speech-to-text translation," *ArXiv preprint*, vol. abs/1612.01744, 2016. [Online]. Available: https://arxiv.org/abs/1612.01744

[296] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly translate foreign speech," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, F. Lacerda, Ed. ISCA, 2017, pp. 2625–2629. [Online]. Available: http://www.isca-speech.org/archive/Interspeech_2017/abstracts/0503.html

[297] T. Etchegoyhen, H. Arzelus, H. Gete, A. Alvarez, I. G. Torre, J. M. Martín-Doñas, A. González-Docasal, and E. B. Fernandez, "Cascade or direct speech translation? a case study," *Applied Sciences*, vol. 12, no. 3, p. 1097, 2022.

[298] H. Inaguma, K. Duh, T. Kawahara, and S. Watanabe, "Multilingual end-to-end speech translation," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU).* IEEE, 2019, pp. 570–577.

[299] M. Gheini, T. Likhomanenko, M. Sperber, and H. Setiawan, "Joint speech transcription and translation: Pseudo-labeling with out-of-distribution data," *ArXiv preprint*, vol. abs/2212.09982, 2022. [Online]. Available: https://arxiv.org/abs/2212.09982

[300] P. Wang, E. Sun, J. Xue, Y. Wu, L. Zhou, Y. Gaur, S. Liu, and J. Li, "Lamassu: Streaming language-agnostic multilingual speech recognition and translation using neural transducers," *ArXiv preprint*, vol. abs/2211.02809, 2022. [Online]. Available: https://arxiv.org/abs/2211.02809

[301] M. A. Di Gangi, R. Cattoni, L. Bentivogli, M. Negri, and M. Turchi, "MuST-C: a Multilingual Speech Translation Corpus," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers).* Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 2012–2017. [Online]. Available: https://aclanthology.org/N19-1202

[302] A. C. Kocabiyikoglu, L. Besacier, and O. Kraif, “Augmenting librispeech with French translations: A multimodal corpus for direct speech translation evaluation,” in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. [Online]. Available: https://aclanthology.org/L18-1001

[303] M. Post, G. Kumar, A. Lopez, D. Karakos, C. Callison-Burch, and S. Khudanpur, “Improved speech-to-text translation with the fisher and callhome Spanish-English speech translation corpus,” in *Proceedings of the 10th International Workshop on Spoken Language Translation: Papers*, Heidelberg, Germany, Dec. 5-6 2013. [Online]. Available: https://aclanthology.org/2013.iwslt-papers.14

[304] D. Raj, L. Lu, Z. Chen, Y. Gaur, and J. Li, “Continuous streaming multi-talker ASR with dual-path transducers,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7317–7321.

[305] G. Kumar, Y. Cao, R. Cotterell, C. Callison-Burch, D. Povey, and S. Khudanpur, “Translations of the callhome Egyptian Arabic corpus for conversational speech translation,” in *Proceedings of the 11th International Workshop on Spoken Language Translation: Papers*, Lake Tahoe, California, Dec. 4-5 2014, pp. 244–248. [Online]. Available: https://aclanthology.org/2014.iwslt-papers.13

[306] M. Zanon Boito, J. Ortega, H. Riguidel, A. Laurent, L. Barrault, F. Bougares, F. Chaabani, H. Nguyen, F. Barbier, S. Gahbiche, and Y. Estève, “ON-TRAC consortium systems for the IWSLT 2022 dialect and low-resource speech translation tasks,” in *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*. Dublin, Ireland (in-person and online): Association for Computational Linguistics, May 2022, pp. 308–318. [Online]. Available: https://aclanthology.org/2022.iwslt-1.28

[307] Y. Peng, K. Kim, F. Wu, B. Yan, S. Arora, W. Chen, J. Tang, S. Shon, P. Sridhar, and S. Watanabe, “A comparative study on e-branchformer vs conformer in speech recognition, translation, and understanding tasks,” *arXiv preprint arXiv:2305.11073*, 2023.

[308] K. Soky, S. Li, M. Mimura, C. Chu, and T. Kawahara, “Leveraging simultaneous translation for enhancing transcription of low-resource language via cross attention mechanism,” *Proc. Interspeech 2022*, pp. 1362–1366, 2022.

[309] K. Deng, S. Watanabe, J. Shi, and S. Arora, “Blockwise streaming transformer for spoken language understanding and simultaneous speech translation,” *arXiv preprint arXiv:2204.08920*, 2022.

[310] J. Xue, P. Wang, J. Li, M. Post, and Y. Gaur, “Large-scale streaming end-to-end speech translation with neural transducers,” *ArXiv preprint*, vol. abs/2204.05352, 2022. [Online]. Available: https://arxiv.org/abs/2204.05352

[311] N. Kanda, J. Wu, Y. Wu, X. Xiao, Z. Meng, X. Wang, Y. Gaur, Z. Chen, J. Li, and T. Yoshioka, “Streaming Multi-Talker ASR with Token-Level Serialized Output Training,” in *Proc. Interspeech 2022*, 2022, pp. 3774–3778.

[312] Z. Huang, D. Raj, P. García, and S. Khudanpur, “Adapting self-supervised models to multi-talker speech recognition using speaker embeddings,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[313] M. Yang, N. Kanda, X. Wang, J. Wu, S. Sivasankaran, Z. Chen, J. Li, and T. Yoshioka, “Simulating realistic speech overlaps improves multi-talker ASR,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10094928

[314] R. Paturi, S. Srinivasan, K. Kirchhoff, and D. Garcia-Romero, “Directed speech separation for automatic speech recognition of long form conversational speech,” in *Proc. Interspeech 2022*, 2022, pp. 5388–5392.

[315] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, “Listen, attend and spell,” *arXiv preprint arXiv:1508.01211*, 2015.

[316] B. Zhang, B. Haddow, and R. Sennrich, “Revisiting end-to-end speech-to-text translation from scratch,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 26 193–26 205.

[317] B. Zhang *et al.*, “Efficient CTC Regularization via Coarse Labels for End-to-End Speech Translation,” in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Dubrovnik, Croatia: Association for Computational Linguistics, 2023, pp. 2264–2276. [Online]. Available: https://aclanthology.org/2023.eacl-main.166

[318] I. Tsiamas, G. I. Gállego, J. A. Fonollosa, and M. R. Costa-jussà, “Shas: Approaching optimal segmentation for end-to-end speech translation,” *ArXiv preprint*, vol. abs/2202.04774, 2022. [Online]. Available: https://arxiv.org/abs/2202.04774

[319] M. Post, “A call for clarity in reporting BLEU scores,” in *Proceedings of the Third Conference on Machine Translation: Research Papers*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 186–191. [Online]. Available: https://aclanthology.org/W18-6319

[320] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, “Scaling vision transformers,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 104–12 113.

[321] T. Q. Nguyen, K. Murray, and D. Chiang, “Data augmentation by concatenation for low-resource translation: A mystery and a solution,” in *Proceedings of the 18th*

International Conference on Spoken Language Translation (IWSLT 2021). Bangkok, Thailand (online): Association for Computational Linguistics, Aug. 2021, pp. 287–293. [Online]. Available: https://aclanthology.org/2021.iwslt-1.33

[322] L. Lupo, M. Dinarelli, and L. Besacier, "Divide and rule: Effective pre-training for context-aware multi-encoder translation models," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 4557–4572. [Online]. Available: https://aclanthology.org/2022.acl-long.312

[323] H. Bredin, R. Yin, *et al.*, "Pyannote.audio: neural building blocks for speaker diarization," in *ICASSP*. IEEE, 2020, pp. 7124–7128.

[324] H. Bredin and A. Laurent, "End-to-end speaker segmentation for overlap-aware resegmentation," in *Proc. Interspeech 2021*, 2021.

[325] N. Blum, S. Lachapelle, and H. Alvestrand, "Webrtc: Real-time communication for the open web platform," *Communications of the ACM*, vol. 64, no. 8, pp. 50–54, 2021.

[326] E. Matusov, G. Leusch, O. Bender, and H. Ney, "Evaluating machine translation output with automatic sentence segmentation," in *Proceedings of the Second International Workshop on Spoken Language Translation*, Pittsburgh, Pennsylvania, USA, Oct. 24-25 2005. [Online]. Available: https://aclanthology.org/2005.iwslt-1.19

[327] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, 2020.

[328] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-End Speech Processing Toolkit," in *Proc. Interspeech 2018*, 2018, pp. 2207–2211.

[329] M. Saberi, O. Khadeer Hussain, and E. Chang, "Past, present and future of contact centers: a literature review," *Business Process Management Journal*, vol. 23, no. 3, pp. 574–597, 2017.

[330] I. Medennikov, M. Korenevsky, T. Prisyach, Y. Khokhlov, M. Korenevskaya, I. Sorokin, T. Timofeeva, A. Mitrofanov, A. Andrusenko, I. Podluzhny, *et al.*, "Target-speaker voice activity detection: a novel approach for multi-speaker diarization in a dinner party scenario," *arXiv preprint arXiv:2005.07272*, 2020.

[331] S.-Y. Chang, R. Prabhavalkar, Y. He, T. N. Sainath, and G. Simko, "Joint endpointing and decoding with end-to-end models," in *ICASSP*. IEEE, 2019, pp. 5626–5630.

[332] T. J. Park, N. Kanda, D. Dimitriadis, K. J. Han, S. Watanabe, and S. Narayanan, "A review of speaker diarization: Recent advances with deep learning," *Computer Speech & Language*, vol. 72, p. 101317, 2022.

[333] J. Li, A. Sun, J. Han, and C. Li, “A survey on deep learning for named entity recognition,” *IEEE transactions on knowledge and data engineering*, vol. 34, no. 1, pp. 50–70, 2020.

[334] Y. Xu, H. Zhao, and Z. Zhang, “Topic-aware multi-turn dialogue modeling,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 16, 2021, pp. 14 176–14 184.

[335] S. Ghannay, A. Caubriere, Y. Esteve, A. Laurent, and E. Morin, “End-to-end named entity extraction from speech,” *arXiv preprint arXiv:1805.12045*, 2018.

[336] Y.-C. Chen, S.-w. Yang, C.-K. Lee, S. See, and H.-y. Lee, “Speech representation learning through self-supervised pretraining and multi-task finetuning,” *arXiv preprint arXiv:2110.09930*, 2021.

[337] S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, *et al.*, “SUPERB: Speech Processing Universal PERformance Benchmark,” in *Proc. Interspeech 2021*, 2021, pp. 1194–1198.

[338] S. Kumar, S. Madikeri, N. Iuliia, E. VILLATORO-TELLO, P. Motlicek, K. P. D. S, S. P. Dubagunta, and A. Ganapathiraju, “Multitask speech recognition and speaker change detection for unknown number of speakers,” in *Proceedings of the 49th IEEE International Conference on Acoustics, Speech, & Signal Processing (ICASSP) 2024*, 2024.

[339] I. Cohn, I. Laish, G. Beryozkin, G. Li, I. Shafran, I. Szpektor, T. Hartman, A. Hassidim, and Y. Matias, “Audio de-identification: A new entity recognition task,” *arXiv preprint arXiv:1903.07037*, 2019.

[340] Y. Wu, S. Maiti, Y. Peng, W. Zhang, C. Li, Y. Wang, X. Wang, S. Watanabe, and R. Song, “Speechcomposer: Unifying multiple speech tasks with prompt composition,” *arXiv preprint arXiv:2401.18045*, 2024.

[341] K.-W. Chang, Y.-K. Wang, H. Shen, I.-t. Kang, W.-C. Tseng, S.-W. Li, and H.-y. Lee, “Speechprompt v2: Prompt tuning for speech classification tasks,” *arXiv preprint arXiv:2303.00733*, 2023.

[342] L. E. Shafey, H. Soltau, and I. Shafran, “Joint speech recognition and speaker diarization via sequence transduction,” *arXiv preprint arXiv:1907.05337*, 2019.

[343] W. Xia, H. Lu, Q. Wang, A. Tripathi, Y. Huang, I. L. Moreno, and H. Sak, “Turn-to-diarize: Online speaker diarization constrained by transformer transducer speaker turn detection,” in *ICASSP*. IEEE, 2022, pp. 8077–8081.

[344] S. Cornell, J.-w. Jung, S. Watanabe, and S. Squartini, “One model to rule them all? towards end-to-end joint speaker diarization and speech recognition,” *arXiv preprint arXiv:2310.01688*, 2023.

[345] H. Yadav, S. Ghosh, Y. Yu, and R. R. Shah, “End-to-end named entity recognition from english speech,” *arXiv preprint arXiv:2005.11184*, 2020.

[346] H. Bredin, C. Barras, *et al.*, “Speaker change detection in broadcast tv using bidirectional long short-term memory networks,” in *Interspeech 2017*. ISCA, 2017.

[347] A. Raux and M. Eskenazi, “Optimizing endpointing thresholds using dialogue features in a spoken dialogue system,” in *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, 2008, pp. 1–10.

[348] H. Bredin, “Pyannote.audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe,” in *24th INTERSPEECH Conference*. ISCA, 2023, pp. 1983–1987.

[349] Bredin, Hervé, “Pyannote.metrics: A Toolkit for Reproducible Evaluation, Diagnostic, and Error Analysis of Speaker Diarization Systems,” in *Proc. Interspeech 2017*. ISCA, 2017, pp. 3587–3591.

[350] G. Kumar, M. Post, D. Povey, and S. Khudanpur, “Some insights from translating conversational telephone speech,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 3231–3235.

[351] Y. Peng, J. Tian, W. Chen, S. Arora, B. Yan, Y. Sudo, M. Shakeel, K. Choi, J. Shi, X. Chang, *et al.*, “OWSM v3. 1: Better and Faster Open Whisper-Style Speech Models based on E-Branchformer,” *arXiv preprint arXiv:2401.16658*, 2024.

[352] Y. Peng, Y. Sudo, M. Shakeel, and S. Watanabe, “OWSM-CTC: An Open Encoder-Only Speech Foundation Model for Speech Recognition, Translation, and Language Identification,” *arXiv preprint arXiv:2402.12654*, 2024.

[353] A. Gu, K. Goel, and C. Re, “Efficiently modeling long sequences with structured state spaces,” in *International Conference on Learning Representations*, 2021.

[354] A. Gu and T. Dao, “Mamba: Linear-time sequence modeling with selective state spaces,” *arXiv preprint arXiv:2312.00752*, 2023.

[355] O. Lieber, B. Lenz, H. Bata, G. Cohen, J. Osin, I. Dalmedigos, E. Safahi, S. Meirom, Y. Belinkov, S. Shalev-Shwartz, *et al.*, “Jamba: A hybrid transformer-mamba language model,” *arXiv preprint arXiv:2403.19887*, 2024.

[356] C. Wang, J. Pino, A. Wu, and J. Gu, “CoVoST: A Diverse Multilingual Speech-To-Text Translation Corpus,” in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 2020, pp. 4197–4203.

A Appendix to Section 6.1, Chapter 6

A.1 Fisher-CALLHOME Data Distribution

In Table A.1, we list the main characteristics for each train, development, and test set of the Fisher-CALLHOME corpora. Blue subsets denote splits with the original segmentation provided by LDC and the authors of the Fisher-CALLHOME translations [303], which is widely used by previous work [98, 296, 307, 298]. The red denote segmentation with MT-MS data (see §6.1 for more details) and orange when VAD is applied on the merged audio stream per conversation. It is worth mentioning that MT-MS segmentation yields fewer samples than VAD segmentation because our “conversations" might contain, noise and silences between consecutive utterances;

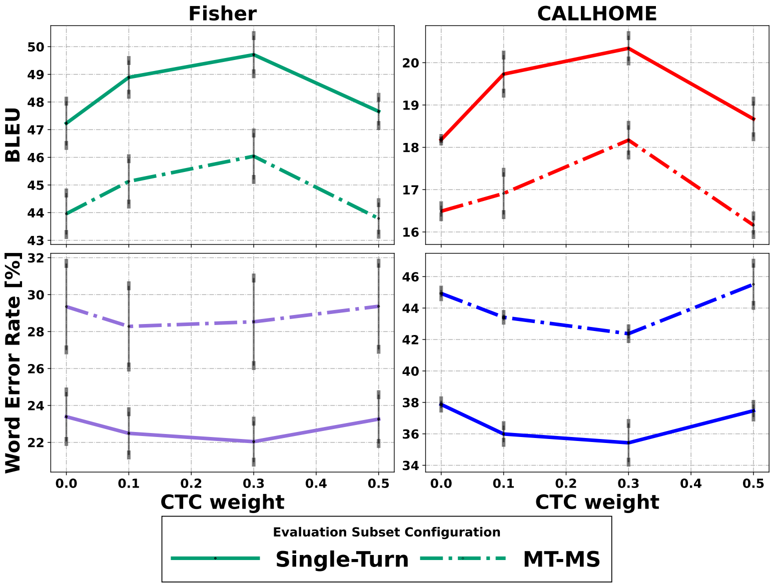


Figure A.1: Ablation of the CTC weight in the overall loss computation and its impact in BLEU and WERs for Fisher and CALLHOME development & evaluation sets. Error bars show the standard deviation between dev/dev2/test sets for Fisher and devset/evlset for CALLHOME. Single-turn and MS-MS results are shown with straight and dashed lines, respectively.

Table A.1: Main characteristics of each train, development and test subset of Fisher and CALLHOME corpora, after pre-processing. Pre-processing includes, segmentation by ground truth metadata, multi-turn & multi-speaker segmentation, or †voice activity detection-based segmentation with SHAS [318] algorithm. Minimum and maximum segment length for SHAS are set to 1 and 30 seconds, respectively.

Development/Test Subset	# samples	Time [hrs]	Speech [hr/%]	Non-speech [hr/%]
Fisher				
train-single-turn	138764	150.60	146.52/(97.29%)	4.08/(2.71%)
train-multi-turn	22051	150.62	149.40/(99.19%)	1.22/(0.81%)
dev-single-turn	3977	3.99	3.88/(97.30%)	0.11/(2.70%)
dev-multi-turn	572	3.99	3.95/(98.87%)	0.05/(1.13%)
dev-resegemented†	867	3.98	3.55/(89.20%)	0.43/(10.80%)
dev2-single-turn	3958	3.92	3.84/(97.94%)	0.08/(2.06%)
dev2-multi-turn	580	3.92	3.90/(99.47%)	0.02/(0.53%)
dev2-resegemented	849	3.92	3.52/(89.81%)	0.40/(10.19%)
test-single-turn	3641	4	3.90/(97.65%)	0.09/(2.35%)
test-multi-turn	583	4	3.97/(99.41%)	0.02/(0.59%)
test-resegemented†	856	3.99	3.61/(90.50%)	0.38/(9.50%)
CALLHOME				
callhome-train-single-turn	15042	16.20	12.65/(78.10%)	3.55/(21.90%)
callhome-train-multi-turn	1905	16.20	13.41/(82.79%)	2.79/(17.21%)
callhome-devtest-single-turn	3956	4	3.19/(79.66%)	0.81/(20.34%)
callhome-devtest-multi-turn	482	4	3.35/(83.63%)	0.66/(16.37%)
callhome-devtest-resegmented†	745	4.00	2.98/(74.41%)	1.02/(25.59%)
callhome-evltest-single-turn	1825	2.71	1.58/(58.29%)	1.13/(41.71%)
callhome-evltest-multi-turn	242	2.71	1.66/(61.41%)	1.04/(38.59%)
callhome-evltest-resegmented†	358	2.71	1.49/(55.06%)	1.22/(44.94%)

A.2 Evaluating Different CTC Weights

In this section, we evaluate different CTC weights for joint ASR & ST training under the STAC-ST framework. We show in Figure A.1 the results for different S-size models trained on the Fisher-CALLHOME corpora. We confirm that BLEU and WER scores achieve the best with a $\lambda = 0.3$, akin to previous work [316].

A.3 Complete Main Evaluation Results on Fisher-CALLHOME

We list complete main results on Fisher-CALLHOME corpora for all the official subsets.

Multi-Turn Segments. Table A.2 lists BLEU scores for all subsets of Fisher-CALLHOME, while Table A.3 lists WER scores.

Table A.2: BLEU scores on each multi-turn dataset for all the official Fisher-CALLHOME development and test subset. AVG lists the average between dev and test sets.

Training Data				**BLEU score (↑)**						
Single-turn		**Multi-turn**		**Fisher**				**CALLHOME**		
ASR	ST	ASR	ST	dev	dev2	test	AVG	devtest	evltest	AVG
	✓		✓	26.2	27.0	28.3	27.2	8.6	8.5	8.5
✓	✓			25.6	27.0	29.3	27.3	8.8	8.9	8.8
		✓	✓	40.2	40.0	41.5	40.5	15.0	14.7	14.8
✓	✓	✓		32.7	32.9	35.6	33.7	10.6	11.7	11.1
✓	✓		✓	42.3	42.5	43.7	42.8	15.2	15.5	15.4
✓	✓	✓	✓	45.1	46.1	46.8	46.0	18.4	17.9	18.2

Table A.3: WERs on each multi-turn dataset for all the official Fisher-CALLHOME development and test subset. AVG lists the average between dev and test sets.

Training Data				**Word Error Rate (↓)**						
Single-turn		**Multi-turn**		**Fisher**				**CALLHOME**		
ASR	ST	ASR	ST	dev	dev2	test	AVG	devtest	evltest	AVG
✓		✓		29.7	30.0	26.1	28.6	44.0	43.5	43.8
✓	✓			45.9	46.6	40.2	44.2	58.0	57.9	58.0
		✓	✓	35.2	35.8	29.4	33.5	51.4	49.9	50.7
✓	✓	✓		29.4	30.0	25.8	28.4	42.9	42.3	42.6
✓	✓		✓	52.8	54.6	44.9	50.8	64.3	68.2	66.3
✓	✓	✓	✓	30.2	29.6	25.8	28.5	42.6	42.1	42.4

Table A.4: BLEU scores on each single-turn dataset for all the official Fisher-CALLHOME development and test subset. AVG lists the average between dev and test sets.

Training Data				**BLEU score (↑)**						
Single-turn		**Multi-turn**		**Fisher**				**CALLHOME**		
ASR	ST	ASR	ST	dev	dev2	test	AVG	devtest	evltest	AVG
	✓		✓	34.1	34.5	34.3	34.3	11.4	11.0	11.2
✓	✓			50.2	51.5	50.0	50.5	21.2	21.2	21.2
		✓	✓	41.1	41.6	41.7	41.4	14.8	14.9	14.8
✓	✓	✓		47.5	48.1	47.1	47.5	18.5	19.2	18.8
✓	✓		✓	47.2	47.7	46.6	47.2	19.4	18.6	19.0
✓	✓	✓	✓	49.6	50.4	49.1	49.7	20.5	20.1	20.3

Single-Turn Segments. For the sake of completeness, we also report the performance of `STAC-ST` on each subset of Fisher-CALLHOME with the default utterance segmentation (single-turn).

Table A.5: WERs on each single-turn dataset for all the official Fisher-CALLHOME development and test subset. AVG lists the average between dev and test sets.

Training Data				**Word Error Rate (↓)**						
Single-turn		**Multi-turn**		**Fisher**				**CALLHOME**		
ASR	ST	ASR	ST	dev	dev2	test	AVG	devtest	evltest	AVG
✓		✓		23.5	22.8	21.0	22.5	35.5	36.3	35.9
✓	✓			22.8	22.2	20.7	21.9	34.0	34.6	34.3
		✓	✓	31.5	31.6	27.9	30.3	48.4	48.4	48.4
✓	✓	✓		23.1	22.5	20.8	22.1	35.2	35.6	35.4
✓	✓		✓	26.0	26.1	23.4	25.2	38.7	39.7	39.2
✓	✓	✓	✓	23.0	22.2	20.8	22.0	34.6	36.3	35.4

Table A.6: Performance of STAC-ST on speaker change detection on the multi-turn dataset for all official Fisher-CALLHOME test sets. Tolerance is ablated from 0.1 up to 1 second.

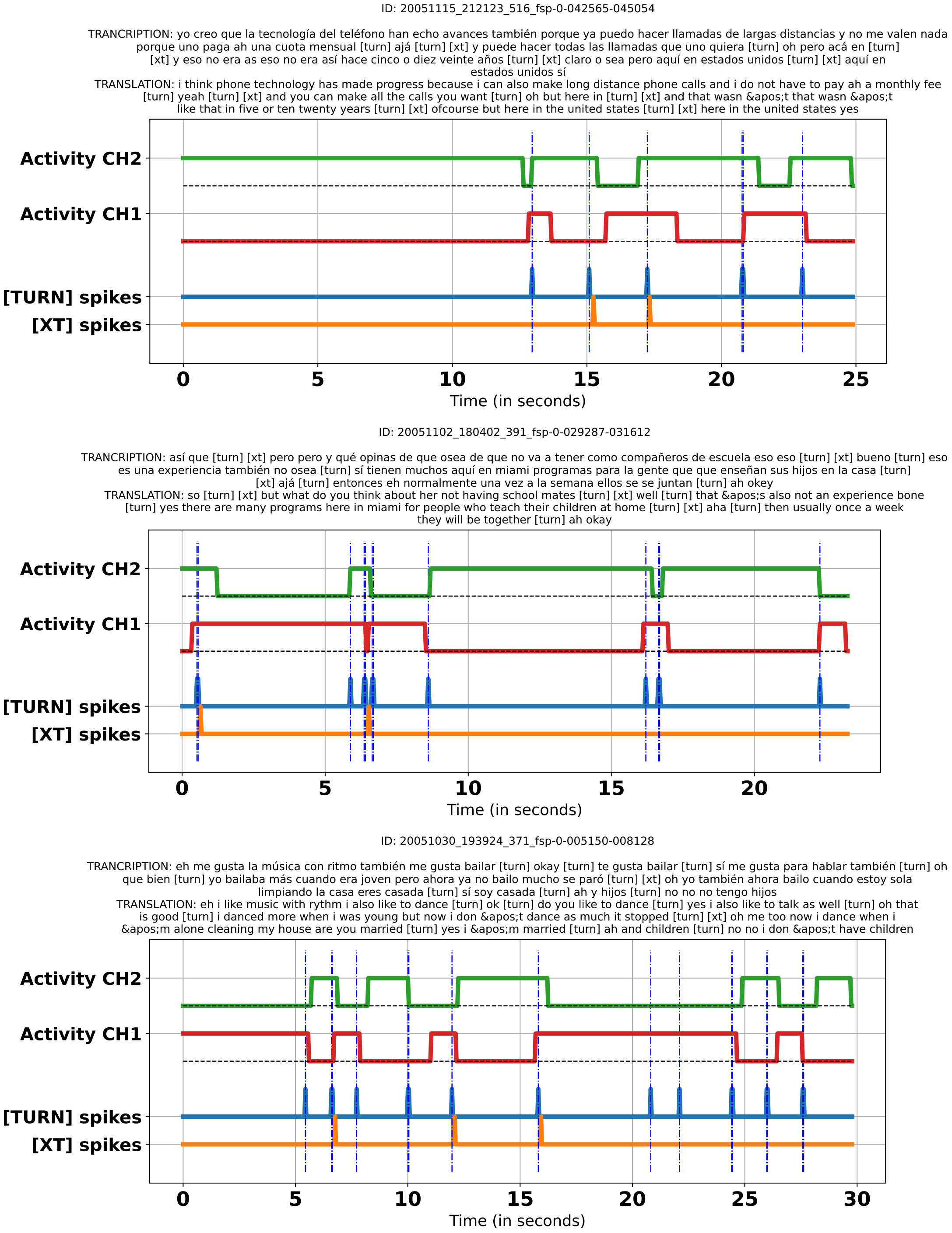
TOL	Fisher			CALLHOME		
(s)	F1	MDR	FAR	F1	MDR	FAR
0.1	58.3	46.2	36.4	67.6	37.5	26.4
0.25	74.9	31.3	17.7	80.6	25.6	12.1
0.5	83.4	23.0	9.0	85.5	20.8	7.2
1	87.3	18.4	6.2	89.3	16.2	4.5

Table A.4 lists the BLEU scores, while Table A.5 list WER scores.

A.4 More Examples and Analysis on Speaker-Turn and Cross-Talk Detection

In Figure A.2, we provide 3 additional examples of ground-truth speaker activities vs. CTC spikes of [TURN] and [XT] task tokens (see §6.1). The title contains the sample ID, transcript and translation together with the [TURN] and [XT] task tokens.

In Table A.6 we evaluate different tolerance values when computing the speaker change detection metrics con both Fisher-CALLHOME test sets. The tolerance (in seconds) allows us to reduce the granularity that we expect in speaker change detection. Giving the fact that STAC-ST is not directly optimized for this task, we note that a value of at least 0.25 is critical to reach acceptable scores – by increasing the tolerance from 0.1 to 0.25 seconds, we see a 22% relative increase in F1 score. Setting it to 0.5 seconds further brings a 10% relative improvement.



ID: 20051115_212123_516_fsp-0-042565-045054

TRANCRIPTION: yo creo que la tecnología del teléfono han echo avances también porque ya puedo hacer llamadas de largas distancias y no me valen nada porque uno paga ah una cuota mensual [turn] ajá [turn] [xt] y puede hacer todas las llamadas que uno quiera [turn] oh pero acá en [turn] [xt] y eso no era as eso no era así hace cinco o diez veinte años [turn] [xt] claro o sea pero aquí en estados unidos [turn] [xt] aquí en estados unidos sí
TRANSLATION: i think phone technology has made progress because i can also make long distance phone calls and i do not have to pay ah a monthly fee [turn] yeah [turn] [xt] and you can make all the calls you want [turn] oh but here in [turn] [xt] and that wasn &apos;t that wasn &apos;t like that in five or ten twenty years [turn] [xt] ofcourse but here in the united states [turn] [xt] here in the united states yes

ID: 20051102_180402_391_fsp-0-029287-031612

TRANCRIPTION: así que [turn] [xt] pero pero y qué opinas de que osea de que no va a tener como compañeros de escuela eso eso [turn] [xt] bueno [turn] eso es una experiencia también no osea [turn] sí tienen muchos aquí en miami programas para la gente que que enseñan sus hijos en la casa [turn] [xt] ajá [turn] entonces eh normalmente una vez a la semana ellos se se juntan [turn] ah okey
TRANSLATION: so [turn] [xt] but what do you think about her not having school mates [turn] [xt] well [turn] that &apos;s also not an experience bone [turn] yes there are many programs here in miami for people who teach their children at home [turn] [xt] aha [turn] then usually once a week they will be together [turn] ah okay

ID: 20051030_193924_371_fsp-0-005150-008128

TRANCRIPTION: eh me gusta la música con ritmo también me gusta bailar [turn] okay [turn] te gusta bailar [turn] sí me gusta para hablar también [turn] oh que bien [turn] yo bailaba más cuando era joven pero ahora ya no bailo mucho se paró [turn] [xt] oh yo también ahora bailo cuando estoy sola limpiando la casa eres casada [turn] sí soy casada [turn] ah y hijos [turn] no no no tengo hijos
TRANSLATION: eh i like music with rythm i also like to dance [turn] ok [turn] do you like to dance [turn] yes i also like to talk as well [turn] oh that is good [turn] i danced more when i was young but now i don &apos;t dance as much it stopped [turn] [xt] oh me too now i dance when i &apos;m alone cleaning my house are you married [turn] yes i &apos;m married [turn] ah and children [turn] no no i don &apos;t have children

Figure A.2: Ground-truth speaker activities and CTC spikes of `[TURN]` and `[XT]` task tokens on three randomly selected Fisher samples. The Tile list the ID (recording, file number, start and end time), the ground truth transcript and translation.

Table A.7: Ablation of the impact of encoding speaker turn and cross-talk information with [TURN] and [XT]. BLEU scores and WERs are listed for multi-turn dataset for all the official Fisher-CALLHOME development and test sets. AVG lists the average between dev and test sets.

Special Tokens	Fisher				CALLHOME		
	dev	dev2	test	AVG	devtest	evltest	AVG
BLEU score (↑)							
N/A	43.4	44.2	45.0	44.2	17.0	16.6	16.8
[TURN]	44.2	44.7	45.2	44.7	17.6	17.6	17.6
[TURN] + [XT]	**45.1**	**46.1**	**46.8**	**46.0**	**18.4**	**17.9**	**18.1**
***Word Error Rate* (↓)**							
N/A	29.9	30.3	26.4	28.9	43.9	43.7	43.6
[TURN]	**29.2**	31.1	**25.8**	28.7	43.2	43.1	43.2
[TURN] + [XT]	30.2	**29.6**	**25.8**	**28.5**	**42.6**	**42.1**	**42.4**

A.5 Complete Ablation Results for [TURN] & [XT] Task Tokens

We provide compete ablation results of adding [TURN] & [XT] task tokens on all the official development and test sets of Fisher-CALLHOME, as listed in Table A.7.

A.6 More Details of VAD-Based Segmentation

With WebRCT, audio is split when 90% of consecutive frames do not include speech. We set the frame length parameter to 30 ms and the aggressiveness parameter to 1 as in [318]. With SHAS, we set 1-30 as the min-max sequence length.

SHAS was trained on monologue corpora with MuST-C [301]. Thus, we perform an additional pre-processing step to minimize the domain mismatch between SHAS and Fisher-CALLHOME. (1) We extract the speech activity boundaries for each audio file from the original metadata. (2) We modify each audio file by masking with 0 all the regions in the signal where there is no speech activity, i.e., setting all the non-speech activity regions to silence. (3) We then use the masked long-form audio files with SHAS. This step decreases the false alarms rate that can be produced by SHAS on noisy segments or between contiguous utterances where there are close-talks. Close-talks are areas where two utterances are too close and the segmentation tools might not generalize well. In order to keep comparable the experimental and evaluation setup, we perform the same pre-processing step when using WebRCT.

Besides SHAS, we also plot the segmentation distribution of WebRCT on the Fisher test set in Figure A.3. WebRCT yields a more reasonable distribution than SHAS. Note that some samples are longer than 30 seconds. We compare different segmentation techniques with two training data configurations in Figure A.4: only **Single**-turn data, i.e., Row-4 in Table 6.2; **Both** single-turn

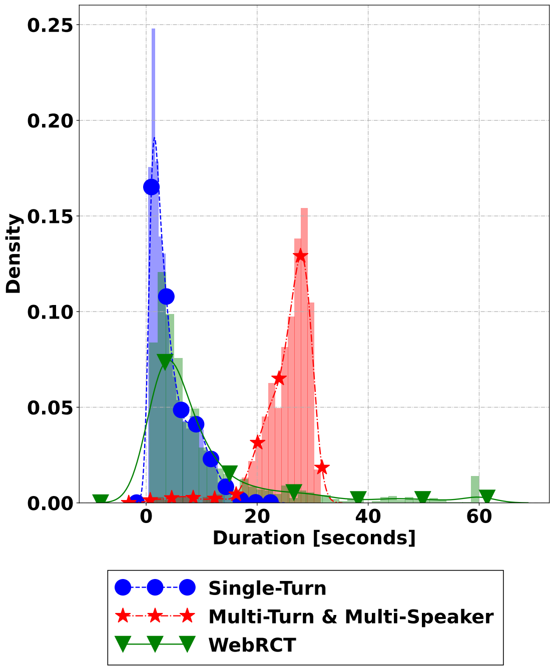


Figure A.3: Data distribution for Fisher test set with different segmentation approaches.

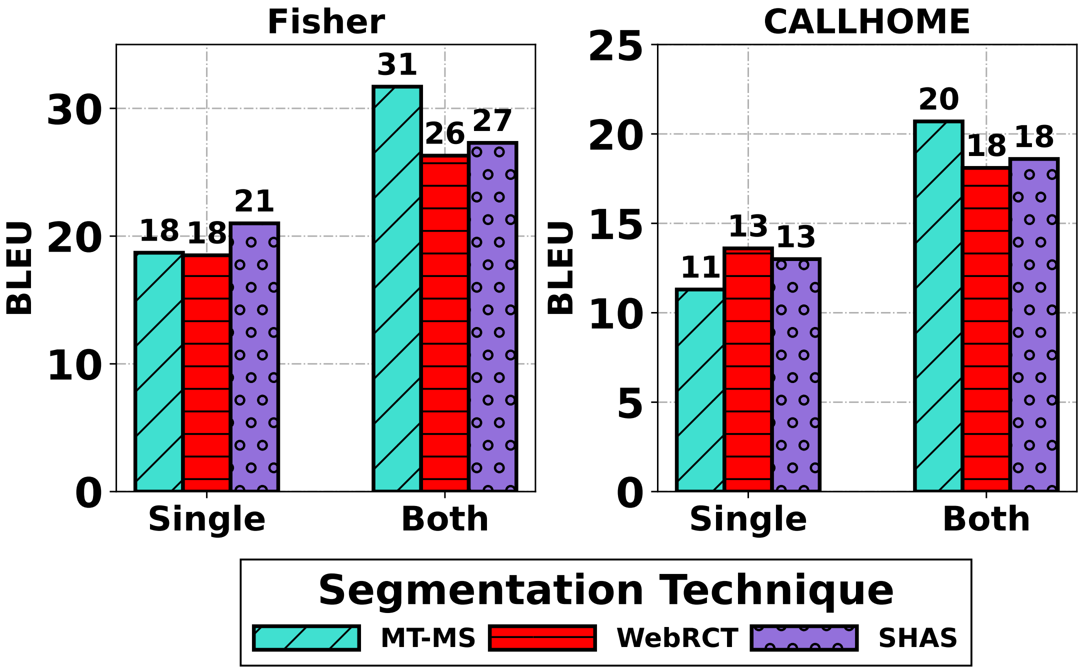


Figure A.4: We compare different segmentation techniques with two training data configurations: only **Single**-turn data and **Both** single-turn and multi-turn data. The bars denote different segmentation techniques for long-form audio, including, MT-MS segmentation (proposed in this work), VAD via WebRCT [325] or SHAS [318].

and multi-turn data, i.e., Row-3 in Table 6.2. Using our proposed configuration, Both, helps all segmentation techniques we tested during inference.

Table A.8: Comparison between Whisper versus scaled STAC-ST using more training data. WER and BLEU scores are reported on the multi-turn dataset for all the official Fisher-CALLHOME development and test subsets. AVG lists the average between dev and test sets.

		Fisher				**CALLHOME**		
Model	**Size (θ)**	dev	dev2	test	AVG	devtest	evltest	AVG
				BLEU score (↑)				
Whisper-tiny	39M	8.1	7.5	11.5	9.0	1.9	2.4	2.2
Whisper-base	74M	27.4	23.7	29.0	26.7	7.3	8.4	7.9
Whisper-small	244M	44.2	44.1	46.7	45.0	19.2	19.2	19.2
Whisper-medium	769M	48.6	47.7	49.2	48.5	22.5	23.1	22.8
STAC-ST (S)	21M	45.1	46.1	46.8	46.0	18.4	17.9	18.2
STAC-ST (M)	86M	48.1	48	49.4	48.5	20.2	20.4	20.3
STAC-ST (L)	298M	48.6	48.9	50.0	49.2	21.0	21.0	21.0
				Word Error Rate (↓)				
Whisper-tiny	39M	51.5	50.1	45.0	48.9	60.3	59.8	60.1
Whisper-base	74M	41.8	42.0	36.7	40.2	50.0	49.2	49.6
Whisper-small	244M	33.9	33.7	29.1	32.2	39.1	37.9	38.5
Whisper-medium	769M	31.3	30.9	28.7	30.3	33.9	32.3	33.1
STAC-ST (S)	21M	30.2	29.6	25.8	28.5	42.6	42.1	42.4
STAC-ST (M)	86M	27.0	28.1	23.8	26.3	40.1	38.3	39.2
STAC-ST (L)	298M	27.9	27.9	23.5	26.4	38.98	38.5	38.7

A.7 Complete Results of Scaled STAC-ST vs. Whisper

We list complete evaluation results of scaled STAC-ST vs. Whisper for the MT-MS Fisher-CALLHOME development and test sets in Table A.8.

A.8 Complete Results of STAC-ST for Single-Turn ST

We list complete evaluation results of STAC-ST vs. prior work for the single-turn Fisher-CALLHOME development and test sets in Table A.9. Note that in Section 6.1, we only list the work that (1) released the Fisher-CALLHOME corpora [303] and (2) the top three models that report both WER and BLEU scores.

Table A.9: Comparison between previous work vs. scaled STAC-ST. WER and BLEU scores are reported on single-turn segments of all the official Fisher-CALLHOME development and test subsets. AVG lists the average between dev and test sets. We list the best BLEU/WER scores for each model from previous work. In some cases, it includes ASR or MT pre-training. †Multilingual model, name convention in [298].

		Fisher				CALLHOME		
Model	**Size** (θ)	dev	dev2	test	AVG	devtest	evltest	AVG
				BLEU score (↑)				
Cas. ASR-MT [303]		-	35.5	-	-	-	11.6	-
Multi-task ASR/ST [296]		48.3	49.1	48.7	48.7	16.8	17.4	17.1
E2E-ST M2Mc† [298]		44.1	45.4	45.2	44.9	16.4	16.2	16.3
EMc2+ASR-PT† [298]		46.3	47.1	46.3	46.6	17.3	17.2	17.3
E2E-ST streaming [309]		47.9	48.2	47.7	47.9	15.5	15.3	15.4
ESPnet [328]		51.8	52.3	50.5	51.5	22.3	21.7	22.0
Whisper-tiny	39M	7.4	5.6	9.0	7.3	2.0	2.2	2.1
Whisper-base	74M	19.1	20.4	25.4	21.6	6.0	6.5	6.2
Whisper-small	244M	45.4	40.7	45.3	43.8	17.5	16.8	17.1
Whisper-medium	769M	51.7	49.2	48.8	49.9	23.5	23.5	23.5
STAC-ST (S)	21M	49.6	50.4	49.1	49.7	20.5	20.1	20.3
STAC-ST (M)	86M	52.0	51.9	52.3	52.1	23.0	22.1	22.6
STAC-ST (L)	298M	52.4	52.8	52.6	52.6	22.7	22.4	22.5
				Word Error Rate (↓)				
SAT-fMLLR [303]		41.3	40.0	36.5	39.3	64.7	65.3	65.0
SAT-SGMM [98]		35.9	34.5	-	-	-	-	-
Multi-task ASR/ST [296]		25.7	25.1	23.2	24.7	44.5	45.3	44.9
E2E-ST M2Ma† [298]		25.6	25.0	22.9	24.5	43.5	44.5	44.0
Joint ASR+MT [308]		22.8	22.3	20.5	21.9	39.5	39.4	39.5
From ESPnet [328]		20.5	20.2	18.7	19.8	37.8	37.6	37.7
Whisper-tiny	39M	50.9	49.9	44.1	48.3	60.5	58.5	59.5
Whisper-base	74M	41.4	39.5	34.8	38.6	49.0	48.7	48.8
Whisper-small	244M	32.2	30.5	28.1	30.2	36.9	36.5	36.7
Whisper-medium	769M	28.3	26.8	25.8	27.0	29.8	29.3	29.6
STAC-ST (S)	21M	23.0	22.2	20.9	22.0	34.6	36.3	35.4
STAC-ST (M)	86M	21.1	20.4	18.9	20.1	30.2	31.4	30.8
STAC-ST (L)	298M	21.0	20.6	18.8	20.1	30.4	31.0	30.7

Table A.10: Main characteristics of MSLT dataset [1] used in our experiments. We list the details for each language pair and for each task, i.e., ASR and ST.

Characteristics	**FR → FR & EN**				**DE → DE & EN**				**EN → EN & DE**			
	ASR		ST		ASR		ST		ASR		ST	
	dev	test	dev	test	dev	test	dev	test	dev	test	dev	test
Nb. of samples [k]	1.5	1.5	1.4	1.5	1.5	1.7	1.5	1.7	2.4	2.4	2.4	2.4
Duration [hr]	2.94	3.02	2.91	3.0	3.32	3.56	3.31	3.56	3.79	3.84	3.79	3.84

A.9 Microsoft Speech Language Translation (MSLT) Corpus Detailed Results

This appendix list the detailed results for the official development and evaluation subsets for the Microsoft Speech Language Translation Corpus (MSLT) [1].[1] MSLT dataset was created from real-life conversations over Skype. The authors provided metadata and manually segmented audio together with translations, i.e., a 3-way dataset. This corpus was part of IWSLT-2016 campaign [1]. In this work we use three different language pairs: EN→DE, DE→EN and, FR→EN. Details for each language pair are listed in Table A.10. Note that, differently from previous work, we list ASR and BLEU scores results on both dev and test sets.

A.9.1 Dataset characteristics

Table A.11 list the results for the DE → DE & EN direction, while Table A.12 and Table A.13, cover the results for EN → EN & DE and FR → FR & EN, respectively.

[1] See: https://www.microsoft.com/en-us/download/details.aspx?id=54689

Table A.11: BLEU scores and WERs for different models trained with CoVoST2 and CommonVoice and evaluated on the DE → DE & EN direction of MSLT corpus. †note that this row denotes a second round of training: we fine-tune the given model on the dev set (+FT-DEV) and evaluate on test. ‡models pre-trained with at least 10k hours of Microsoft data.

Language Pair	**Size (θ)**	**DE → DE & EN**			
		ASR		ST	
		dev	test	dev	test
Baselines					
LAMASSU-UNI‡ [300]		-	-	-	18.7
DE → DE/EN	21M	45.8	45.6	4.0	4.1
↪ +FT-DEV†	21M	-	28.4	-	14.3
DE → DE/EN	86M	40.4	40.1	3.3	3.0
↪ +FT-DEV†	86M	-	27.1	-	15.3
ALL→ALL	86M	36.1	35.4	8.4	8
↪ +FT-DEV†	86M	-	25.2	-	18.5
ALL→ALL	298M	33.3	33.0	9.3	8.9
↪ +FT-DEV†	298M	-	22.6	-	19.9

Table A.12: BLEU scores and WERs for different models trained with CoVoST2 and CommonVoice and evaluated on the EN → EN & DE direction of MSLT corpus. †note that this row denotes a second round of training: we fine-tune the given model on the dev set (+FT-DEV) and evaluate on test. ‡models pre-trained with at least 10k hours of Microsoft data.

Language Pair	**Size (θ)**	**EN → EN & DE**			
		ASR		ST	
		dev	test	dev	test
Baselines					
TT‡ [310]		-	-	-	30.7
LAMASSU-UNI‡ [300]		-	-	-	20.0
EN → EN/DE	21M	54.4	58.0	5.6	5.3
↪ +FT-DEV†	21M	-	29.9	-	13.8
EN → EN/DE	86M	53.4	52.6	8	8.4
↪ +FT-DEV†	86M	-	26.6	-	17.3
ALL→ALL	86M	35.8	37.5	8.9	8.6
↪ +FT-DEV†	86M	-	24.8	-	17.0
ALL→ALL	298M	32.5	34.1	8.8	8.3
↪ +FT-DEV†	298M	-	22.2	-	19.3

Table A.13: BLEU scores and WERs for different models trained with CoVoST2 and CommonVoice and evaluated on the FR → FR & EN direction of MSLT corpus. †note that this row denotes a second round of training: we fine-tune the given model on the dev set (+FT-DEV) and evaluate on test.

Language Pair	**Size (θ)**	**FR → FR & EN**			
		ASR		ST	
		dev	test	dev	test
FR → FR/EN	21M	57.5	54.8	10.7	12.2
↪ +FT-DEV†	21M	-	32.2	-	22.2
FR → FR/EN	86M	52.8	50.4	12.5	13.6
↪ +FT-DEV†	86M	-	29.1	-	24.0
ALL→ALL	86M	45.8	44.5	15.0	15.9
↪ +FT-DEV†	86M	-	27.5	-	25.7
ALL→ALL	298M	43.0	41.7	14.8	16.3
↪ +FT-DEV†	298M	-	25.0	-	27.5

A.10 STAC-ST on CoVoST2 & CommonVoice

In this work, we also intend to demonstrate that STAC-ST also generalizes to non-conversational ST and ASR. This appendix supports that STAC-ST, (1) generalizes to two well-known non-conversational ST & ASR benchmarks. (2) generalizes to language pairs not covered by Fisher-CALLHOME corpora, including two additional XX→EN directions (DE/FR) and EN→DE direction. (3) can be scaled up in both, data and model size. The train/dev/test sets sizes for this ablation are listed in Table A.14.

Table A.14: CoVoST2 and CommonVoice dataset splits used in our work. We list the number of samples (#) and cumulative hours (Hr.) per each subset. †this experiment joins all the available train datasets per each language pair, during evaluation, we test on each single-language pair.

Language Pair	**CommonVoice [97]**						**CoVoST2 [2]**					
	TRAIN		DEV		TEST		TRAIN		DEV		TEST	
	#	Hr.	#	Hr.	#	Hr.	#	Hr.	#	Hr.	#	Hr.
FR → FR & EN	507k	731	16k	25	16k	26	205k	264	14k	21	14k	23
DE → DE & EN	537k	855	16k	27	15k	27	127k	184	13k	20	13k	21
EN → EN & DE	1012k	1602	16k	27	16k	26	287k	428	15k	26	15k	24
ES → ES & EN	277k	406	15k	26	15k	26	78k	113	13k	21	13k	22
ALL → ALL†	2333k		-	-	-	-	697k	-	-	-	-	-

Baseline Results Table A.15 list the results for each proposed language direction from CoVoST2 and CommonVoice against baselines from previous work. Note that in practice, all our models are bilingual because they are optimized for both, ASR and ST.

A.10.1 Scaling Up STAC-ST

We evaluate STAC-ST on four +CoVoST2+CV language directions (see §6.1). We confirm that BLEU and WER scores improves as we scale up STAC-ST model size. This result is not surprising as it has already being proven in computer vision [320], speech [78] and NLP [327]. Yet, it is fundamental to verify that STAC-ST can be scaled up in standard ST and ASR benchmarks. Similarly, Table A.9 shows that bottom most STAC-ST model, with 298M parameters, beats strong baselines based on Whisper, further proving our system as a good fit to jointly model ASR & ST.

Table A.15: WERs and BLEU scores on different language directions of CoVoST2 [2] corpus. Numbers denote performance on the test set.

Nb. Parameters	**DE → DE & EN**		**EN → EN & DE**		**FR → FR & EN**		**ES → ES & EN**	
θ (M)	WER (↓)	BLEU (↑)	WER (↓)	BLEU (↑	WER (↓)	BLEU (↑	WER (↓)	BLEU (↑)
Baselines								
Whisper [188]†	13.0	25.3	14.5	-	22.7	27.3	10.3	33.0
XLSR model [50]‡	-	26.7	-	23.6	-	32.9	-	34.1
21M	11.8	19.7	21.6	14.0	15.1	29.2	10.1	29.1
86M	8.9	20.5	15.7	20.7	12.8	31.8	8.0	32.6
298M	7.8	21.4	14.5	21.0	13.3	25.2	7.5	33.0
ALL–>ALL 298M	7.6	27.5	14.6	20.7	11.4	34.0	6.5	35.8

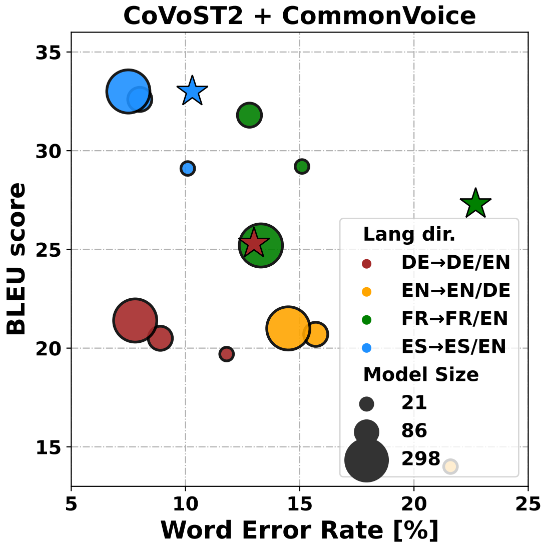


Figure A.5: WERs and BLEU scores on four different language directions of CoVoST2 [2, 356] corpus. Note that the top left systems denote better overall performance, i.e., higher BLEU and lower WER. Star markers denote performance by Whisper-small. Note that there is no reference for EN→EN/DE as it only runs on XX→EN language pair.

Juan Pablo Zuluaga-Gomez

Speech & Audio Processing Research Group
Idiap Research Institute
Rue Marconi 19, 1920 Martigny
(+41) 027 721 77 11
Website: juanpzuluaga.github.io
Google Scholar: https://scholar.google.com/citations?user=_9_Ja2MAAAAJ&hl=en
Emails: juan.zuluaga@eu4m.eu juan-pablo.zuluaga@idiap.ch

EDUCATION

PhD Candidate, Electrical Engineering & Computer Science
École polytechnique fédérale de Lausanne, Vaud, Switzerland *January 2020 - July 2024*
THESIS - Low-Resource Speech Recognition and Understanding for Challenging Applications
Supervisor: Petr Motlicek, PhD.

Master of Science, Mechatronic Engineering
Universidad de Oviedo, Spain & ENSMM, France *September 2017 - September 2019*
THESIS - Breast Cancer Diagnosis Based on Computer Vision
Score 89/100
Supervisor: Noureddine Zerhouni, PhD.

Bachelor of Science, Mechatronic Engineering
Universidad Autonoma del Caribe, Barranquilla, Colombia *January 2011 - December 2015*
THESIS - Precordial Signal Detection System by Seismocardiography
Score 91/100
Supervisor: Pablo Bonaveri, PhD.

PROFESSIONAL EXPERIENCE

Apple, AI/ML Team, Cambridge, MA, USA July – September 2023
Machine Learning Engineer—Internship

- Working on discriminative training of language models to improve automatic speech recognition (ASR) performance on tail named-entity data.
- Transformer-based language modeling for production-level ASR systems.

Amazon, Amazon Web Services (AWS), Seattle, WA, USA April – July 2023
Applied Scientist—Internship

- Member of the AWS AI Transcribe & Translate teams.
- Research on dual speech-to-text Translation (ST) and Transcription (ASR) for conversational speech.
- Serialized output training (conditioned with special tokens, akin to Whisper) for robust multilingual ST and ASR.
- Our system is aware of speaker turns and overlapped speech, improving BLEU and WER performance.

Idiap Research Institute, VS, Switzerland January 2020 - July 2024
Doctor of Philosophy (Ph.D.) - Candidate

- Automatic speech recognition (ASR) for air traffic control (ATC): ATCO2 EU-H2020.
- Implemented innovative semi-supervised techniques for ASR in air-traffic control (low resource task).
- Led the integration of natural language processing (NLP) techniques. 50% improvement in named-entity recognition from ASR transcripts (breakthrough).
- Developed systems for speaker role and speaker change detection based on ASR transcripts.

- Participated at several venues: EMNLP, INTERSPEECH, ICASSP, OSN Symposium (8 conf.).
- Implemented a streaming ASR system for ATC communications: collaboration with industrial partners.
- Participation on industrial projects: spoken language understanding (use case: call-centers).

Research Institute Femto-ST, Besancon, France February 2019 - October 2019
Master of Science Thesis

- Participated: SBRA-"Smart BRA" project, financed by INTERREG (France-Suisse)
- Developed a system for breast cancer diagnosis based on thermal images.
- Early research in multi-modal techniques (vision & signal) for breast cancer diagnosis.
- Published two journal papers.
- Master Thesis: Breast Cancer Diagnosis Using Machine Learning.

Universidad Autonoma del Caribe, Barranquilla, Colombia
Mechatronic Research Group Member UAC September 2014 -

- Participation in national and international events (7), co-authorship in publications (4), 2 patents.
- Active member of the GIIM research group of mechatronic, as a senior research student for three consecutive years and then an active member.

Bachelor Student January 2011 – December 2015

- Main topics covered: automatic control, electronics, mechanical systems, robotics, nanotechnology, machine learning and computer science.
- Research on Titanium dioxide (TiO_2) for wastewater decontaminaton: work as student on the GIIM research group in Mechatronics.
- Thesis: developed a system to capture and analyze precordial signals by Seismocardiography and signal processing.

PUBLICATIONS (JOURNAL, PEER REVIEWED)

7. **Zuluaga-Gomez, Juan**, Veselý, K., Szöke, I., Motlicek, P., et al. (2022). ATCO2 corpus: A Large-Scale Dataset for Research on Automatic Speech Recognition and Natural Language Understanding of Air Traffic Control Communications. *Under review at Data-centric Machine Learning Research Journal, arXiv preprint arXiv:2211.04054*.

6. **Zuluaga-Gomez, Juan**, Prasad, A., Nigmatulina, I., Motlicek, P., & Kleinert, M. (2023). A virtual simulation-pilot agent for training of air traffic controllers. *Aerospace*, *10*(5). https://doi.org/10.3390/aerospace10050490.

5. **Zuluaga-Gomez, Juan**, Nigmatulina, I., Prasad, A., Motlicek, P., Khalil, D., Madikeri, S., Tart, A., Szoke, I., Lenders, V., Rigault, M., & Choukri, K. (2023). Lessons learned in transcribing 5000 h of air traffic control communications for robust automatic speech understanding. *Aerospace*, *10*(10). https://doi.org/10.3390/aerospace10100898.

4. Khalil, D., Prasad, A., Motlicek, P., **Zuluaga-Gomez, Juan**, Nigmatulina, I., Madikeri, S., & Schuepbach, C. (2023). An Automatic Speaker Clustering Pipeline for the Air Traffic Communication Domain. *Aerospace*, *10*(10). https://doi.org/10.3390/aerospace10100876.

3. Ahrenhold, N., Helmke, H., Mühlhausen, T., Ohneiser, O., Kleinert, M., Ehr, H., Klamert, L., & **Zuluaga-Gomez, Juan**. (2023). Validating Automatic Speech Recognition and Understanding for Pre-Filling Radar Labels – Increasing Safety While Reducing Air Traffic Controllers' Workload. *Aerospace*, *10*(6). https://doi.org/10.3390/aerospace10060538.

2. Zhan, Q., Xie, X., Hu, C., **Zuluaga-Gomez, Juan**, Wang, J., & Cheng, H. (2021). Domain-adversarial based model with phonological knowledge for cross-lingual speech recognition. *Electronics*, *10*(24). https://doi.org/10.3390/electronics10243172.

1. **Zuluaga-Gomez, Juan** et al. (2021a). A CNN-based methodology for breast cancer diagnosis using thermal images. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, *9*(2).

PUBLICATIONS (JOURNAL, REVIEW PAPER, PEER REVIEWED)

2. **Zuluaga-Gomez, Juan**, Bonaveri, P., Zuluaga, D., et al. (2020). Techniques for water disinfection, decontamination, and desalinization: A review. *Desalination And Water Treatment.*

1. **Zuluaga-Gomez, Juan** et al. (2019). A survey of breast cancer screening techniques: Thermography and electrical impedance tomography. *Journal of medical engineering & technology, 43*(5).

PUBLICATIONS (CONFERENCE, PEER REVIEWED)

22. **Zuluaga-Gomez, Juan**, Huang, Z., Niu, X., Paturi, R., Srinivasan, S., Mathur, P., Thompson, B., & Federico, M. (2023). End-to-End Single-Channel Speaker-Turn Aware Conversational Speech Translation. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (main).* https://arxiv.org/abs/2311.00697.

21. **Zuluaga-Gomez, Juan**, Ahmed, S., Visockas, D., & Subakan, C. (2023). CommonAccent: Exploring Large Acoustic Pretrained Models for Accent Classification Based on Common Voice. *Proc. Interspeech 2023* ***[Nominated as best student paper award]****.*

20. *Mai, F., ***Zuluaga-Gomez, Juan**, Parcollet, T., & Motlicek, P. (2023). HyperConformer: Multi-head HyperMixer for Efficient Speech Recognition. *Proc. Interspeech 2023.* ***[Equal contribution]****.*

19. Nigmatulina, I., Madikeri, S., Villatoro-Tello, E., Motliček, P., **Zuluaga-Gomez, Juan**, Pandia, K., & Ganapathiraju, A. (2022). Implementing contextual biasing in GPU decoder for online ASR. *Proc. Interspeech 2023.*

18. Villatoro-Tello, E., Madikeri, S., **Zuluaga-Gomez, Juan**, Sharma, B., Sarfjoo, S. S., Nigmatulina, I., Motlicek, P., Ivanov, A. V., & Ganapathiraju, A. (2023). Effectiveness of text, acoustic, and lattice-based representations in spoken language understanding tasks. *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5.

17. Helmke, H., Kleinert, M., Ahrenhold, N., Ehr, H., Mühlhausen, T., Ohneiser, O., Klamert, L., Motlicek, P., Prasad, A., **Zuluaga Gomez, Juan**, et al. (2023). Automatic Speech Recognition and Understanding for Radar Label Maintenance Support Increases Safety and Reduces Air Traffic Controllers' Workload. *Proceedings of the 15th USA/Europe Air Traffic Management Research and Development Seminar (ATM2023), Savannah, GA, USA, 17.* ***[Best paper award—Human Factors Track]****.*

16. **Zuluaga-Gomez, Juan**, Prasad, A., Nigmatulina, I., et al. (2022). How Does Pre-trained Wav2Vec 2.0 Perform on Domain Shifted ASR? An Extensive Benchmark on Air Traffic Control Communications. *2022 IEEE Spoken Language Technology Workshop (SLT), Doha, Qatar.*

15. **Zuluaga-Gomez, Juan**, Sarfjoo, S. S. et al. (2022). BERTraffic: BERT-based Joint Speaker Role and Speaker Change Detection for Air Traffic Control Communications. *2022 IEEE Spoken Language Technology Workshop (SLT), Doha, Qatar.*

14. *Prasad, A., ***Zuluaga-Gomez, Juan** et al. (2022). Speech and Natural Language Processing Technologies for Pseudo-Pilot Simulator. *12th SESAR Innovation Days.*

13. Helmke, H., Ondřej, K., Shetty, S., Arilíusson, H., Simiganoschi, T., Kleinert, M., Ohneiser, O., Ehr, H., **Zuluaga-Gomez, Juan**, & Smrz, P. (2022). Readback Error Detection by Automatic Speech Recognition and Understanding – Results of HAAWAII Project for Isavia's Enroute Airspace. *12th SESAR Innovation Days.*

12. Prasad, A., **Zuluaga-Gomez, Juan**, Motlicek, P., et al. (2022). Grammar Based Speaker Role Identification for Air Traffic Control Speech Recognition. *12th SESAR Innovation Days.*

11. Nigmatulina, I., **Zuluaga-Gomez, Juan**, Prasad, A., et al. (2022). A two-step approach to leverage contextual data: speech recognition in air-traffic communications. *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6282–6286.

10. **Zuluaga-Gomez, Juan** et al. (2021b). Contextual Semi-Supervised Learning: An Approach to Leverage Air-Surveillance and Untranscribed ATC Data in ASR Systems. *Proc. Interspeech 2021.*

9. Kocour, M., Veselý, K., Szoke, I., Kesiraju, S., **Zuluaga-Gomez, Juan**, Blatt, A., et al. (2021). Automatic Processing Pipeline for Collecting and Annotating Air-Traffic Voice Communication Data. *Engineering Proceedings*, *13*(1), 8.

8. Kocour, M., Veselý, K., Blatt, A., **Juan Zuluaga-Gomez**, et al. (2021). Boosting of Contextual Information in ASR for Air-Traffic Call-Sign Recognition. *Proc. Interspeech 2021.*

7. **Zuluaga-Gomez, Juan**, Veselý, K. et al. (2020). Automatic call sign detection: Matching air surveillance data with air traffic spoken communications. *Multidisciplinary Digital Publishing Institute Proceedings*, *59*(1), 14.

6. **Zuluaga-Gomez, Juan**, Motlicek, P. et al. (2020). Automatic Speech Recognition Benchmark for Air-Traffic Communications. *Interspeech*, 2297–2301. https://doi.org/10.21437/Interspeech.2020-2173.

5. Ma, J., Shang, P., Lu, C., Meraghni, S., Benaggoune, K., **Zuluaga-Gomez, Juan**, Zerhouni, N., Devalland, C., & Al Masry, Z. (2019). A portable breast cancer detection system based on smartphone with infrared camera. *Vibroengineering Procedia*, *26*, 57–63.

4. Bonaveri, P., Barrios, M., & **Zuluaga-Gomez, Juan**. (2017). Diseño y Construcción de un Sistema Basado en Acelerometría para la Captación y Análisis en Matlab de Señales Precordiales usando Sismocardiografía 3D. *Memorias del Congreso Nacional de Ingeniería Biomédica*, *2*(1), 153–156.

3. **Zuluaga-Gomez, Juan** et al. (2018). Aprendizaje orientado a proyectos integradores y perfeccionamiento del trabajo en equipo: Caso máster erasmus mundus en ingenieria mecatronica. *XXVI Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas.*

2. **Zuluaga-Gomez, Juan**, & Bonaveri, P. (2016). Sistema para la detección de señales precordiales mediante sismocardiografia. *Prospectiva*, *14*(1), 89–95.

1. Corredor, S., Valbuena, M., **Zuluaga-Gomez, Juan**, & Barrios, M. (2014). Design and Construction a measurer of total body water, fat mass and fat free mass using LabVIEW. *2014 III International Congress of Engineering Mechatronics and Automation (CIIMA)*, 1–4.

PUBLICATIONS (PRE-PRINT)

8. Ravanelli, M., Parcollet, T., Moumen, A., de Langen, S., Subakan, C., Plantinga, P., Wang, Y., Mousavi, P., Libera, L. D., Ploujnikov, A., Paissan, F., Borra, D., Zaiem, S., Zhao, Z., Zhang, S., Karakasidis, G., Yeh, S.-L., Rouhe, A., Braun, R., . . . others. [**Submitted to JMLR**]. (2024). Open-Source Conversational AI with SpeechBrain 1.0. https://arxiv.org/abs/2407.00463.

7. **Zuluaga-Gomez, Juan**, Kumar, S. et al. (2024). Improved Transducer Streaming ASR With Attention Sinks. *To be Submitted to ARR 2024 (long paper).*

6. *Nigmatulina, I., ***Zuluaga-Gomez, Juan** et al. (2024). Fast Streaming Transducer ASR Prototyping via Knowledge Distillation with Whisper. *Submitted to EMNLP 2024 (long paper).* ***[Equal contribution].***

5. Nigmatulina, I., **Zuluaga-Gomez, Juan** et al. (2024). Improved contextual adaptation with an external n-gram language model for Transducer-based ASR. *Submitted to INTERSPEECH 2024.*

4. Kumar, S., Madikeri, S., **Zuluaga-Gomez, Juan**, et al. (2024b). XLSR-Transducer: Streaming ASR for Self-Supervised Pretrained Models. *Submitted to INTERSPEECH 2024.*

3. Kumar, S., Madikeri, S., **Zuluaga-Gomez, Juan**, et al. (2024a). TokenVerse: Unifying Speech and NLP Tasks via Transducer-based ASR. *Submitted to INTERSPEECH 2024.*

2. Nigmatulina, I., Braun, R., **Zuluaga-Gomez, Juan**, & Motlicek, P. (2021). Improving callsign recognition with air-surveillance data in air-traffic communication. *arXiv preprint arXiv:2108.12156.*

1. Madikeri, S., Tong, S., **Zuluaga-Gomez, Juan**, Vyas, A., Motlicek, P., & Bourlard, H. (2020). Pkwrap: A pytorch package for lf-mmi training of acoustic models. *arXiv preprint arXiv:2010.03466.*

PUBLICATIONS (BOOK/BOOK CHAPTER)

1. **Zuluaga-Gomez, J** et al. (2017). Tratamiento de aguas residuales mediante el proceso de fotocatalisis con dioxido de titanio (tio2). In U. A. del Caribe (Ed.). Uniautonoma, ISBN: 9789585431010.

WORKSHOPS (CONFERENCE, PEER REVIEWED)

2. Burdisso, S., **Zuluaga-Gomez, Juan**, Fajcik, M., et al. (2022). IDIAPers @ causal news corpus 2022: Causal relation identification using a few-shot and prompt-based fine-tuning of language models. *The 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE @ EMNLP 2022).*

1. Fajcik, M., Singh, M., **Zuluaga-Gomez, Juan**, et al. (2022). Idiapers @ causal news corpus 2022: Extracting cause-effect-signal triplets via pre-trained autoregressive language model. *The 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE @ EMNLP 2022).*

PARTICIPATION IN CONFERENCES

- IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Year: 2022, 2023, 2024
- Interspeech. Year: 2020, 2021, 2023
- OpenSky Network (OSN) Symposium. Year: 2021, 2020
- XXVI CUIEET National Congress, Spain, 2018
- XVIII National and XII International Research Meeting - Colciencias, Colombia, 2015
- XII Departmental meeting of Research, Colombia, 2015
- Biomedical Engineering National Congress, Mexico, 2015
- IV International Mechatronics and Automation Congress, Colombia, 2015
- III International Mechatronics and Automation Congress, Colombia, 2014
- II International Mechatronics and Automation Congress, Colombia, 2013

REVIEWER IN CONFERENCES/JOURNALS

- Journal: IEEE Transactions on Audio, Speech and Language Processing, 2023
- Conference: Interspeech, 2023, 2024
- Conference: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2022, 2023, 2024
- Conference: EMNLP (@CASE workshop), 2022

RESEARCH PROJECTS

- ATCO2 EU-funded Horizon 2020 project. https://www.atco2.org/Website.
- HAAWAII EU-funded Horizon 2020 project. https://www.haawaii.de/wp/Website.
- Minerva CQ company: research on spoken language understanding and robust ASR.
- Uniphore: research on large-scale speech pseudo-labeling with foundational speech models for training streaming ASR models (Transducers).

PROGRAMMING SKILLS

- Experienced in Bash scripting and Git (Github/Gitlab).
- Experienced in Python, PyTorch, Numpy, Google Colab and Jupyter Notebooks.
- Speech Recognition toolkit: Kaldi, k2/Icefall, SpeechBrain, HuggingFace & ESPNet.
- Natural Language Processing toolkit: PyTorch, HuggingFace.
- Parallel experimentation with SGE, SLURM, and Weight & Biases.

SOFTWARE

- **Icefall (k2) & SpeechBrain**: Python/PyTorch based libraries for automatic speech recognition modeling.
- **PkWrap**: Python library for LF-MMI training of acoustic models for automatic speech recognition with Pytorch, https://github.com/idiap/pkwrap
- **MLX-Apple**: MLX AI/DL framework for Apple-silicon. LLMs/Speech.
- **Wav2Vec 2.0 for air traffic control communications**, https://github.com/idiap/w2v2-air-traffic
- **BerTraffic:** Bert-based text-only speaker Diarization, https://github.com/idiap/bert-text-diarization-atc
- **CommonAccent:** accent ID with CommonVoice, https://github.com/JuanPZuluaga/accent-recog-slt2022
- **HyperConfomer:** efficient automatic speech recognition with Conformer and HyperMixer, https://github.com/speechbrain/speechbrain/blob/develop/speechbrain/nnet/hypermixing.py
- **ATCO2:** a 5000-hour corpus for research on Automatic Speech Recognition and Understanding of Air Traffic Control communications. Baselines and code in: https://github.com/idiap/atco2-corpus

TEACHING

Teaching Assistant:

- Deep Learning Course, EE-559, EPFL (Prof. François Fleuret), Spring 2022

TALKS

- Paper presentation. Unsupersived speech recognition (abs, pdf). At *ECCS Seminar: Advanced Topics in Machine Learning*, 2022.
- Keynote: An introduction to speech-based technologies for Natural Language Processing applications. *Mexican NLP Summer School 2021*, Ciudad de Mexico, Mexico, 2021

AWARD, NOTABLE ACHIEVEMENT

1. "CommonAccent: Exploring Large Acoustic Pretrained Models for Accent Classification Based on Common Voice" paper nominated as best student paper award at Interspeech 2023
2. Ranked 2nd and 3rd place at OLR-2021 challenge task 3 & 4, 2021
3. Scholarship: Erasmus Mundus - European Union, EACEA, 2017
4. Scholarship: to attend XVI World Summit of Nobel Peace Laureates in Colombia, 2017
5. Scholarship: to attend XV World Summit of Nobel Peace Laureates in Spain, 2015
6. Distinction: rank obtained during bachelor studies: ranked 1st out of 7, 2015
7. Distinction: six distinctions and scholarships (GPA during bachelor studies), 2012-2015
8. Scholarship: by DAAD (Germany), visiting student September-October, 2014

LANGUAGE SKILLS

- Spanish: Native
- English: Bilingual Proficiency
- French: Limited Working Proficiency

MEMBERSHIPS

- Member of the International Speech Communication Association (ISCA), since 2020
- Graduate Student Member of the Institute of Electrical and Electronics Engineers (IEEE), since 2022

PATENTS

Device for Cardiac Signals Detection, ***Granted*** September 2019

- Seismocardiography system for Cardiac Signals Detection
- Registration number: NC16-175508
- Phase: Granted 2020
- Financing partners: Commerce Chamber of Barranquilla - CIENTECH

Robot for Martial Arts Training - RobPam, ***Granted*** October 2020

- Robotic humanoid to practice martial arts
- Registration number: NC201-0007622
- Phase: Granted 2020
- Financing partners: Commerce Chamber of Barranquilla - CIENTECH

SUPERVISION ACTIVITIES

Universidad Autonoma del Caribe, Barranquilla, Colombia

Mechatronic Engineering Undergraduate Program October 2016

- Development of a biomedical instrument and mobile APP for cardiac signals (SCG) and pulse oximetry monitoring in older people. **Students**: Cristhian Escalona, Dario Garcia.
- Development of a biomedical system to capture, process and visualize impedance cardiographm and electrocardiogram signals in a web page. **Student**: Juan Villalobos, Daniel Castaneda.

INTERESTS

Cooking, Hiking and Traveling, Reading (recommended books), Coffee Brewing.