Leveraging Untranscribed Data for End-to-End Speech and Callsign Recognition in Air-Traffic Communication

Petr Motlicek
Idiap Research Institute, Switzerland
Brno University of Technology, Czech Republic
petr.motlicek@idiap.ch

Shashi Kumar
Idiap Research Institute, Switzerland
EPFL, Switzerland
shashi.kumar@idiap.ch

Driss Khalil
Idiap Research Institute, Switzerland
driss.khalil@idiap.ch

Amrutha Prasad
Idiap Research Institute, Switzerland
Brno University of Technology, Czech Republic
amrutha.prasad@idiap.ch

Christof Schuepbach
Armasuisse Science and Technology, Thun, Switzerland

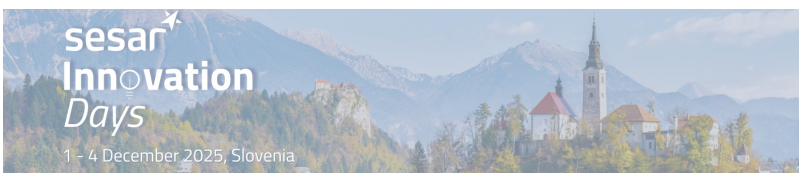
***Abstract*—Accurate Automatic Speech Recognition (ASR) and callsign recognition in Air Traffic Control (ATC) are vital for safety, yet conventional two-step systems rely on large amounts of manually transcribed data, which is both costly and limited. This paper introduces a practical alternative using TokenVerse, a unified end-to-end model trained under a dual-task framework and enhanced through semi-supervised learning. Our main contribution shows that the model can jointly learn callsign boundaries and speech recognition, improving performance on both tasks simultaneously. Additionally, by generating pseudo-labels for 500 hours of unlabeled audio, we substantially expand the effective training data. Experiments across multiple in-domain and out-of-domain ATC datasets demonstrate that the TokenVerse framework achieves state-of-the-art performance in both ASR and callsign detection, surpassing cascaded pipelines built on modern architectures (including Kaldi, XLSR/wav2vec 2.0, Zipformer, and Whisper). This work provides a robust and scalable foundation for deploying and continuously refining high-accuracy ATC systems in real-world settings where labeled data is inherently scarce. The end-to-end architecture is also relatively compact (approximately 317M parameters), making it well suited for real-time, low-latency deployment.**

***Keywords*—automatic speech recognition, semi-supervised learning, air traffic control.**

I. Introduction

There has been a significant progress in automating (i.e., supporting human) speech communication between pilots and Air Traffic Control Officers (ATCOs) by using Machine Learning (ML) technologies, specifically Automatic Speech Recognition (ASR). Although the technology has been validated by users for less-risky deployment such as for Air Traffic Control (ATC) simulators [1], their integration into operational use has not yet been done especially due to issues related to highly-risky applications. In addition to tackling security issues related to general Machine Learning (ML) technologies before their deployment for decision-making, there is still room to increase the accuracies of ASR. Specifically, in the case of ATC, the most critical error, whether caused by a human or a machine, is the incorrect identification of an aircraft (i.e., task of Named Entity Recognition (NER) to be turned into detection of callsign). Incorrectly detecting callsigns (i.e., unique identifiers for aircrafts of which the first part is an abbreviation of 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., RYR1RK) affects other aspects of ATC communication between controllers and pilots, such as errors in issuing specific commands to pilots. Although the decision-making is (and will surely be in near future) fully dependent on the commands issued by the voice (due to its naturalness especially in cases of high workload, etc.), human ATCOs also heavily use the contextual information (specifically radar data which identifies which aircrafts are in the given zone at a certain time point, and thus ATCOs communicate only with a limited number of callsigns). 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 air space and the automatically recognized command (from ATC communication) is invalid. Therefore, contextual information coming from the surveillance (radar) data allows adjusting system predictions that can significantly increase its accuracy. Current solutions considered to supporting humans in ATC communication use cascaded pipeline, combining ASR (speech-to-text) with Natural Language Understanding (NLU) modules [2], [3], where ASR outputs the word-level transcriptions which are subsequently tagged according to the meaning and eventually mapped to the higher-level concepts. Depending on the availability of data, NLU tagging and mapping is either solved using rule-based systems [4], [5], or

This work was supported by Armasuisse Science and Technology.



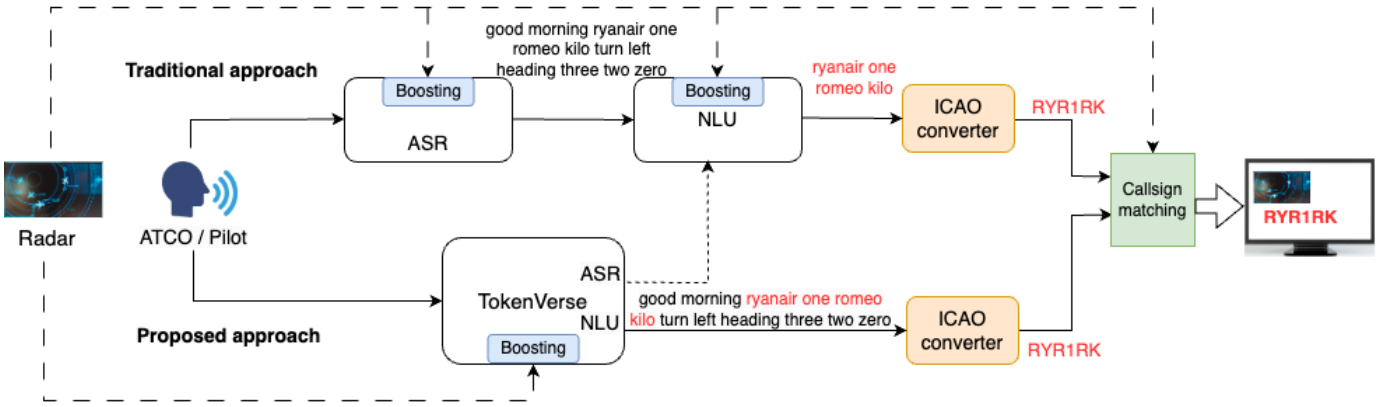


Figure 1. Overview of the traditional and our proposed speech recognition and callsign detection approach. The top part shows the two-step approach commonly employed in callsign extraction. In this the input audio is passed to the ASR system to generate the hypothesis, further used as an input to a rule-based or data driven NLU module. The bottom part shows our proposed method that uses a single system – TokenVerse – generating both the hypothesis text and recognized entity (the callsign). The callsign detection accuracy can further be improved by combining both the output from the external NLU module and TokenVerse.

by using pre-traing Large Language Models (LLMs) adapted to the ATC scenario [6]. This paper proposes the use of multi-task Transducer-based modeling approach through a TokenVerse [7] concept, allowing to train a unified model in end-to-end fashion. The single model is designed to handle multiple tasks by integrating task-specific tokens into the reference text during ASR model training, streamlining the inference and eliminating the need for separate NLU module for subsequent entity recognition (i.e., word tagging). Unlike the multi-head based multitasking approaches proposed in many previous works [8], [9], TokenVerse approach can generate task-specific tokens directly within the ASR hypotheses. Specifically for our use-case, the Transducer architecture allows to attain text and speech alignment for each output token, i.e., we can directly detect and recognize callsigns in the acoustic space. Eventually, simple combination of TokenVerse output with external NLU module can further increase callsign detection accuracies.

Furthermore, as with other ML-based applications, ATC faces challenges due to limited human-annotated data for model training. As the second contribution of this paper, we demonstrate that the proposed TokenVerse approach effectively leverages weakly labeled data, such as speech that is automatically transcribed and annotated by baseline ASR and NLU modules. This establishes TokenVerse as an ideal solution for integration and iterative updates in real-world ATC deployment.

II. Callsign Recognition

The problem defined in this paper can be split into two tasks: (i) use of standard ASR where the input speech sequence is converted into the output word transcriptions, and (ii) NER (Callsign recognition), where the input text sequence is tagged by meaning of each word. Traditional cascaded approach combines ASR and NLU modules to recognize ATC communication and to extract higher-level concepts including callsigns, commands, and other parameters. As visualised in Figure 1, the proposed TokenVerse solution is capable of performing both tasks in end-to-end fashion.

Human ATC operators are also supported by the contextual information available from the radar, including a compressed form of callsigns, i.e., standardized phraseology format of International Civil Aviation Organization (ICAO) [10] (see Figure 2). To introduce the contextual knowledge into the ML system (ASR, NLU), list of available callsigns are expanded to word sequences. The task is not that simple as the compressed form of callsign allows more than one possible realisation in the ATCO/pilot speech: For instance, DLH5KX can be expanded as ‘hansa five kilo x-ray’ or ‘lufthansa five kilo x-ray’, etc. As we can not say which particular expansion is true for an uttered callsign, it is important to take all expansion variants into account.

III. Data and Experimental Setup

As the paper presents and evaluates the two-task ATC speech and entity recognition system, data from ATC domain are used for its development and evaluation. Specifically for the data, we distinguish between an adaptation set—used to train models from scratch or to fine-tune pre-trained ML models—and an evaluation set, which is used to assess the entire technological pipeline encompassing both speech-to-text and callsign detection tasks. To clarify, this paper focuses on callsign detection (a binary task that determines whether a callsign is correctly identified or not), rather than full callsign recognition.

Additionally, we utilize popular (publicly available) ASR architectures pre-trained on large speech datasets (i.e., XLSR (wav2vec 2.0) and Whisper-based models). Further details on the specific model versions are provided in the following sections.

A. Adaptation set

Manually transcribed data: To train or adapt the ASR systems, we use manually labeled data comprising ATC datasets.

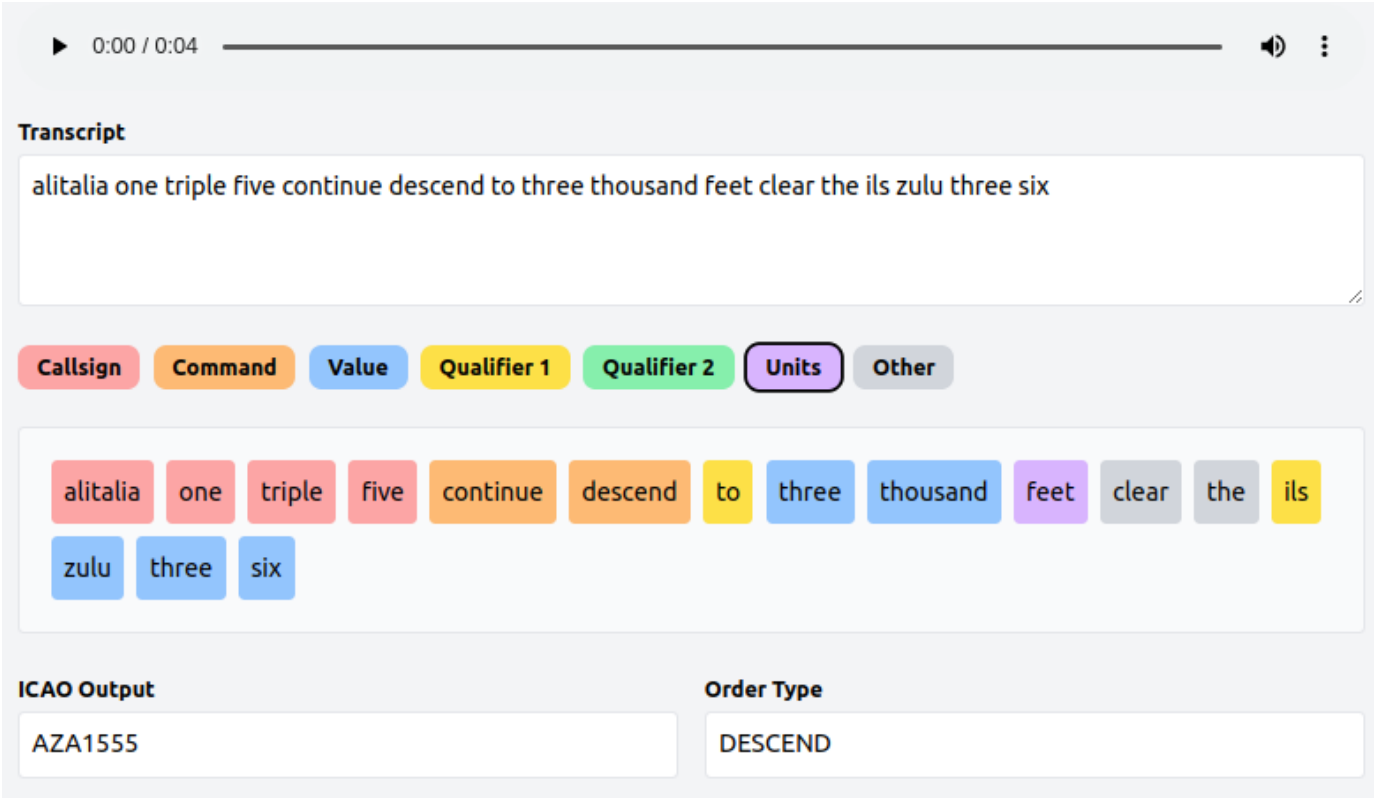


Figure 2. Visualisation of the callsign recognition process: ATCO or pilot transmits the speech, the callsign is recognized and converted into ICAO format, highlighted on the screen.

Detailed description of the datasets are provided in [11]. The total duration of this set is 190 h and consists of approximately 158 k ATC (pilot-ATCO) utterances.

Automatically transcribed data: To further improve the ASR (including TokenVerse) models, we utilize 500 h of untranscribed in-domain ATC data for training. The data has been reported as part of HAAWAII project[1].

Contextual (radar) data is not used in any way for training, as this data either was not available (besides small sub-set supported by radar auxiliary data).

B. Evaluation data

MALORCA (in-domain): the test set offers good quality recordings directly collected at ANSP side (i.e., telephone quality speech with SNR usually above 20dB) as part of the MALORCA project [12], [13][2]. It includes only ATCO speech. Both manually corrected word transcripts as well as ICAO format of callsigns are available. Additionally, for each ATC utterance, we have access to radar data, which contained an average of 5.65 ICAO callsigns per each individual utterance. Data from MALORCA project is also part of adaptation set.

HAAWAII (in-domain): the test set is collected under HAAWAII project with the data coming from the approach (airport). This data is relatively high-quality, similar to MALORCA. Both ATCO and pilot recordings are collected (word transcripts and ICAO format of callsigns). Additionally, for each speech utterance, we have access to radar data, which contains an average of 51.96 ICAO callsigns per utterance. Data from HAAWAII project is also part of adaptation set.

ATCO2 (out-of-domain): This is publicly available data collected as part of ATCO2 project[3]. We use 1 h test set from both ATCOs and pilots. Both manually corrected word transcripts as well as ICAO format of callsigns are available. Compared to MALORCA and HAAWAII, this data is collected using cheap HW and thus the quality (in terms of SNR) is significantly degraded (often SNR $\sim$ 0dB). In this case, we did not have access to radar data. Instead, we used the unique ICAO callsigns from the ground truth as radar data for all utterances, resulting in 355 ICAO callsigns per utterance being used.

IV. Technologies

A. ASR

The following ASR approaches are described here, either considered state-of-the-art or widely used in the past for ATC:

Kaldi [14]: Hybrid based DNN-HMM system is trained from scratch using Kaldi [14] toolkit using adaptation data. MFCCs

[1]https://www.haawaii.de/wp/

[2]From the 'standard' MALORCA test sets [13], only utterances with the available radar data are selected.

[3]https://www.atco2.org/data. There is more than 5000+ hours of speech as publicly available for research and industry.

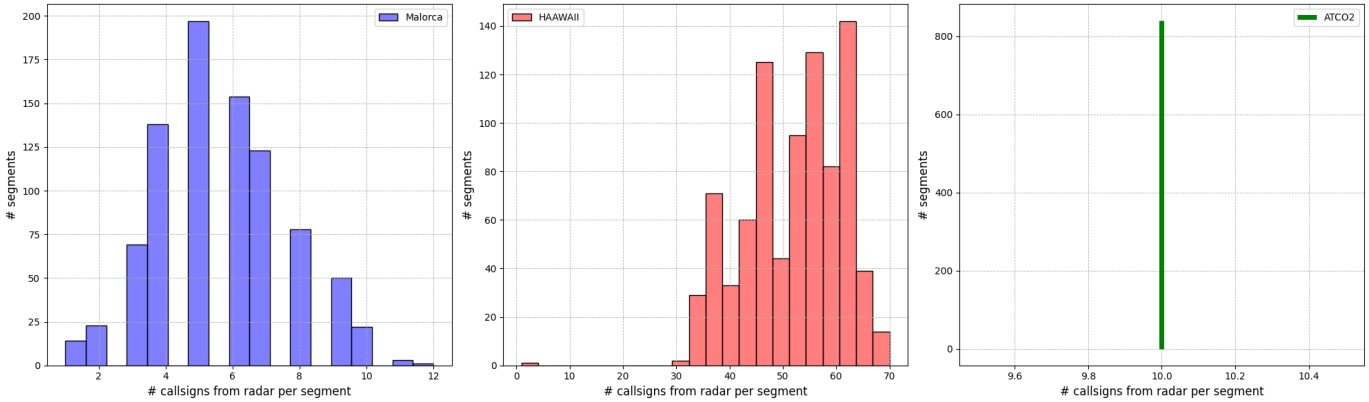


Figure 3. Overview of the callsigns obtained from radar data for different datasets (MALORCA, HAAWAII, ATCO2) used as auxiliary information for Boosting. The x-axis represents the number of callsigns available per utterance, and the y-axis shows how many utterances fall into each count.

and i-vectors features (augmented by 3-fold speed perturbation and one third frame sub-sampling) are modeled using the standard chain training employing Lattice-free MMI (LF-MMI) [15]. The acoustic model uses the conventional biphone convolutional neural network (CNN) [16] + TDNN-F [17] architecture. The language model is 3-gram model trained on the same data as the acoustic model.

XLSR [18]: It is a multilingual model trained in self-supervised fashion based on the wav2vec 2.0 architecture [19]. It is pre-trained using unlabeled speech from 53 languages with a total of 56k hours. The 300M parameter model is used which has 24 transformer layers. We use the LF-MMI adaptation of the pretrained model proposed in [20] for acoustic model training. The espresso [21] toolkit with fairseq [22] is used for finetuning the XLSR model on adaptation data. The model is fine-tuned for 14'000 steps which is 7 epochs with a learning rate of 3e-5. In addition to the pretrained parameters, three layers of factorized TDNN (TDNN-F) [17] are added with a learning rate factor of 20 using the implementation from Pkwrap [23]. The same language model from the above kaldi system is used.

Whisper [24]: We fine-tune the transformer-based encoder-decoder general-purpose model that is trained for various speech processing tasks. Each task is represented by a special token that is jointly predicted by the decoder. We use the 770M parameter 'medium' version of this model trained on 680k hours and consists of 48 transformer layers and is multilingual. The k2/Icefall framework [4] is used for finetuning Whisper medium using adaptation data. The model is fine-tuned for 1 epoch with a learning rate of 1e-5 and label smoothing loss with smoothing factor of 0.1. The features are masked with SpecAug during training [25]. We also investigated the prompting mechanisms to improve the performance following [26], [27]. Recently, the Whisper model has also been presented as an ASR system that leverages boosting capabilities (via its prompting mechanism) within the ATC domain [28].

Zipformer [29]: We use a Transducer architecture specifically designed for ASR. It extends the Transformer architecture with multi-scale encoder blocks, reuses attention computations to reduce overhead, and incorporates custom normalization and activation functions, reducing computational cost while preserving strong recognition performance. The model used[5] is pre-trained with the Gigaspeech [30] dataset, which contains 10k hours of English recordings, and was fine-tuned using adaptation data for 30 epochs with a learning rate of 5e-4 using K2/Icefall toolkit.

B. NLU

Following NLU-based models are used in this paper:

Spacy - rule-based entity-detector: Spacy (a widely used open-source library[6]) is deployed as it offers efficient and scalable tools for text processing and entity extraction. Given the structured nature of callsigns (composed of airline identifiers, numerical sequences, and ATC-specific phonetic words (e.g., "Alfa," "Bravo," "Yankee")), a rule-based approach is well suited for their detection. In this work, we leverage Spacy's EntityRuler, which enables the integration of predefined token-based patterns for precise callsign recognition.

Data-driven entity-detector: We fine-tune a BERT model [31] – `BERT-base-uncased` – with 110 million parameters to the NER task. We use ≈15k utterances (i.e., sentences) from the HAAWAII data which is then augmented by replacing the callsigns with many possible variations. Additionally, we further expand the dataset by augmenting commands and values, resulting in a total of 1 million sentences after augmentation. This is based on the entity parser described in [1]. Overall, BERT-based model did not offer better performance than Spacy, and thus was not further used.

ICAO converter: Our system expands ICAO callsigns into their corresponding phonetic words using a rule-based approach, supported by a comprehensive list of 7'742 airlines. The process begins by identifying the airline based on the first

[4]https://github.com/k2-fsa/icefall

[5]https://huggingface.co/yfyeung/icefall-asr-gigaspeech-zipformer-2023-10-17

[6]https://spacy.io/

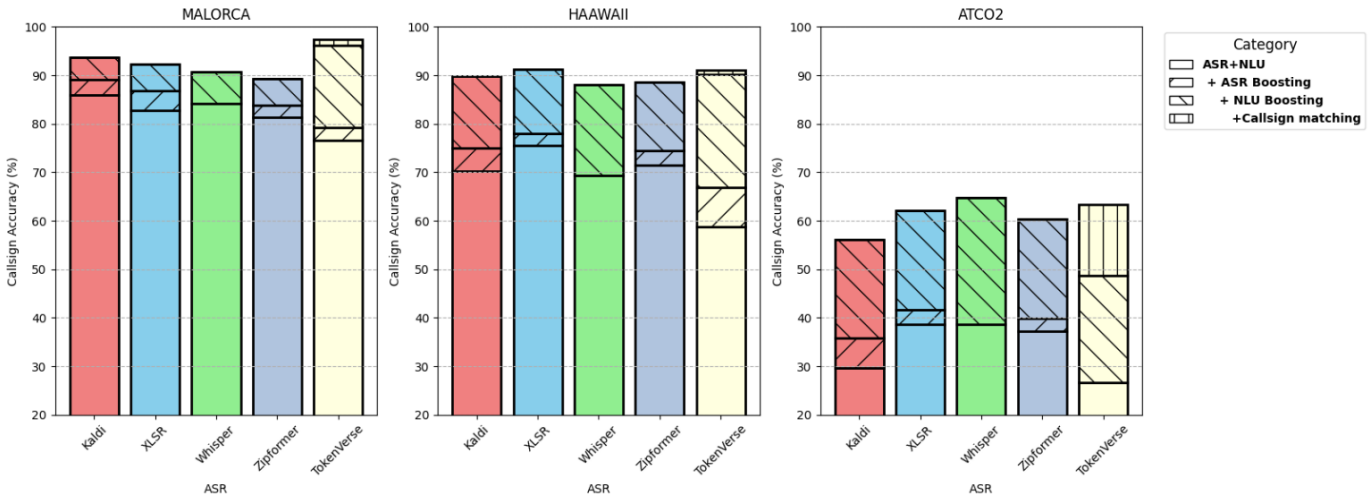


Figure 4. Obtained callsign detection accuracies for proposed ASR system pre-trained on manually transcribed adaptation data. Specifications: (i) ASR+NLU: benchmark approach where input speech is converted into the callsign, (ii) +ASR Boosting: ASR module is boosted by radar data, (iii) +NLU Boosting: both ASR and NLU are boosted by radar data, (iv) +Callsign matching: specifically for TokenVerse, the TokenVerse callsign recognition output is matched with the separated NLU module to further improve callsign detection accuracy.

TABLE I. TEST SETS SUMMARY (NUMBER OF UTTERANCES (UTT), NUMBER OF UTTERANCES WITH NO CALLSIGNS, DURATION (HOURS:MINUTES:SECONDS), NUMBER OF UNIQUE CALLSIGNS, AND AVERAGE NUMBER OF CALLSIGNS PER UTTERANCE (AVAILABLE FROM RADAR)).

Test set	No of utt	Utt with No Callsign	Duration	Unique Callsigns	Avg Csgn per Utt (Radar)
MALORCA	836	103	01:18:14	162	5.65
HAAWAII	852	70	00:48:12	79	51.96
ATCO2	841	86	00:57:17	355	355

three characters of the callsign which correspond to the ICAO airline code. Once the airline is identified, the system retrieves the corresponding callword and expands the rest of the callsign using standard phonetic words. For callsigns that do not match either an airline, the system defaults to a purely phonetic expansion, translating each character (letter or number) into its corresponding word in the ICAO alphabet.

C. TokenVerse

In our work, we leverage the TokenVerse framework to jointly train ASR and callsign entity recognizer . This is achieved by augmenting the training text with special task tokens, enabling the model to recognize and align callsigns within transcriptions. Specifically, we introduce a [CALLSIGN] token before each callsign and a [/CALLSIGN] token after it, allowing the model to explicitly learn callsign boundaries. To generate tokens in adaptation data data, we use Spacy base model as a callsign extractor. The model is trained using an XLSR-Transducer architecture [32] with a SentencePiece [33] tokenizer, ensuring that task-specific tokens are preserved as single subword. During inference, beam search is applied to generate hypotheses, and callsigns are extracted based on the predicted task tokens, where words between the opening and closing tokens are identified as callsigns.

XLSR-Transducer ASR architecture has also an advantage (over other end-to-end types) that it enables streaming capabilities. The work in [32] investigated different attention masking patterns in the self-attention computation of transformer layers within the XLSR-53 model.

D. Boosting

The ASR and NLU modules are further contextualised using available radar data during inference time:

ASR boosting: There are three boosting techniques for enhancing ASR: Shallow Fusion (SF) [34], hot-word boosting [35] and G-boosting [36]. SF employs an n-gram language model (LM) that can be constructed either from contextual data or from training transcripts to estimate the statistics of relevant word sequences during decoding. However, this approach is not used in this paper Instead, this paper applies hot-word boosting and G-boosting. Hot-word boosting is applied to both the Zipformer and TokenVerse systems by incorporating auxiliary information about the potential list of active callsigns during decoding. This approach combines the end-to-end model's posterior probabilities with the auxiliary callsign scores, thereby increasing the likelihood that the relevant callsigns are selected in the final output. On the other hand, G-boosting is used for the Kaldi and XLSR systems, where the n-gram LM (G.fst) is modified by boosting target callsigns before decoding. We used word-level representations

TABLE II. ASR RESULTS (IN WER (%)) OF MODELS WITHOUT USE OF ANY RADAR (CONTEXTUAL DATA). ASR RESULTS FOR WHISPER, ZIPFORMER AND TOKENVERSE MODELS ARE EXTENDED FOR THE CASE OF SSL TRAINING (↑ WER INCREASED, ↓ WER DECREASED). WE ALSO PRESENT NUMBER OF PARAMETERS FOR EACH MODEL.

Model	#Params	MALORCA	HAAWAII	ATCO2
Kaldi	30M	3.6	6.5	29.1
XLSR	320M	3.7	5.1	17.0
Whisper	770M	4.0	6.9	18.6
+ SSL	idem	↑4.4	↑7.7	↑20.0
Zipformer	66M	5.3	6.5	17.6
+ SSL	idem	↓4.4	↓5.6	↓17.1
TokenVerse	317M	5.2	8.1	19.1
+ SSL	idem	↑5.4	↓7.9	↓18.1

of the unique ICAO callsign lists derived from radar data for both the MALORCA and HAAWAII datasets. For ATCO2, however, we generated a simulated auxiliary list of callsigns (consisting of 10 distractor callsigns plus the ground-truth callsign for each utterance) since radar data was not always available.

Figure 3 provides an overview of the number of callsigns used to boost ASR performance for each speech utterance across the datasets.

NLU boosting: NLU boosting improves ICAO callsign recognition by expanding all callsigns available from the radar into multiple word forms and comparing them with the callsign generated by ASR using Levenshtein distance. This method increases the flexibility of matching callsigns, especially in cases where the input data is noisy or incomplete.

E. Callsign matching

TokenVerse naturally allows combining its output with Spacy model. This is carried out through two different approaches: (i) when NLU boosting is applied (i.e., method checks if the callsigns from either TokenVerse or Spacy are present in a reference radar dataset), and (ii) when it is not (i.e., the boosted and non-boosted ICAO callsign outputs are compared using Levenshtein distance; if both ICAOs are identical, one of them is selected).

F. SSL

To improve ASR performance, we leverage Semi-Supervised Learning (SSL) [37] by utilizing 500 hours of untranscribed ATC data in addition to our 190 hours of manually annotated data. We use the Zipformer ASR model to generate pseudo-labels for these recordings. The auxiliary ATC recordings can be automatically pre-selected (i.e., by filtering large set of available speech data) using methods such as those presented here [38], [39].

Additionally, Spacy is used to generate callsigns from the pseudo-labeled transcripts for TokenVerse model training.

V. RESULTS

The paper considers two primary tasks important for the ATC: ASR (speech as input → word transcription as output) and NER (word transcription as input → callsign represented by ICAO format as output). ASR is evaluated using Word-Error-Rate (WER (%)) metric, while callsign detection is measured as a detection problem (Accuracy (%)).

TABLE III. COMPARISON OF CALLSIGN DETECTION ACCURACY (IN %) OF BASELINE XLSR+SPACY APPROACH WITH TOKENVERSE+CALLSIGN MATCHING PIPELINE (ASR AND NLU BOOSTING APPLIED).

Model	MALORCA	HAAWAII	ATCO2
XLSR + Spacy	92.2	91.2	62.2
TokenVerse + Matching	97.6	90.8	63.5

ASR results are summarized in Table II for all 3 test sets. While all ASR systems perform well on the in-domain datasets (MALORCA and HAAWAII), the Kaldi model shows a substantial performance drop on the out-of-domain ATCO2 test set. Notably, a compact Zipformer architecture trained from scratch on the adaptation data can match the performance of the much larger XLSR and Whisper models. Additional one-shot SSL training provides systematic improvements for Zipformer and TokenVerse.

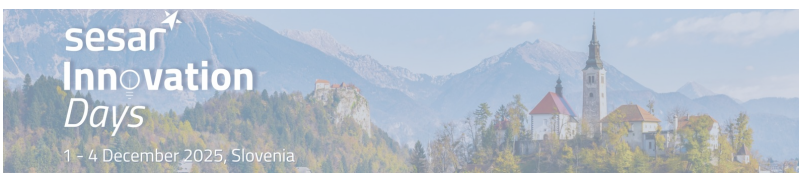
Callsign detection accuracies for proposed ASR pre-trained on manually transcribed adaptation data and further trained on untranscribed data (using SSL) are summarized in Figures 4 and 5, respectively (also highlighted in Table III). End-to-end TokenVerse ASR performs on par with cascaded ASR architectures (Kaldi, XLSR, Whisper, and Zipformer ASR followed by Spacy NLU), achieving the highest callsign detection accuracy of 97.6% for MALORCA dataset, while remaining competitive on the HAAWAII dataset. Callsign matching between TokenVerse and Spacy outputs shows further improvement on the ATCO2 dataset. This is expected as the end-to-end TokenVerse performs considerably worse on out-of-domain noisy ATCO2 data. Leveraging the output from the rule-based Spacy model significantly enhances overall performance.

VI. CONCLUSIONS AND FUTURE WORK

This paper presents the TokenVerse ASR architecture applied to the ATC domain, showcasing its capability to perform both ASR and callsign detection within a single unified framework. TokenVerse can reliably identify incomplete callsigns, which can subsequently be refined through NLU-based boosting. Its integration with rule-based NLU systems (e.g., Spacy) further enhances performance by leveraging predefined linguistic rules. The TokenVerse framework is compared with conventional ASR systems, namely Kaldi and Zipformer ASR models trained from scratch, as well as XLSR and Whisper ASR models pre-trained on large multilingual datasets, each combined with a downstream callsign extraction module.

The study also emphasizes the impact of SSL, demonstrating that TokenVerse is suitable for operational use as it can further improve through learning with pseudo-labeled data. To the best of our knowledge, the presented ASR and callsign detection results are the highest achieved on these data.

Future experiments will incorporate the updated TokenVerse++ model, recently introduced in [40]. This new architecture adds learnable vectors to the acoustic embedding



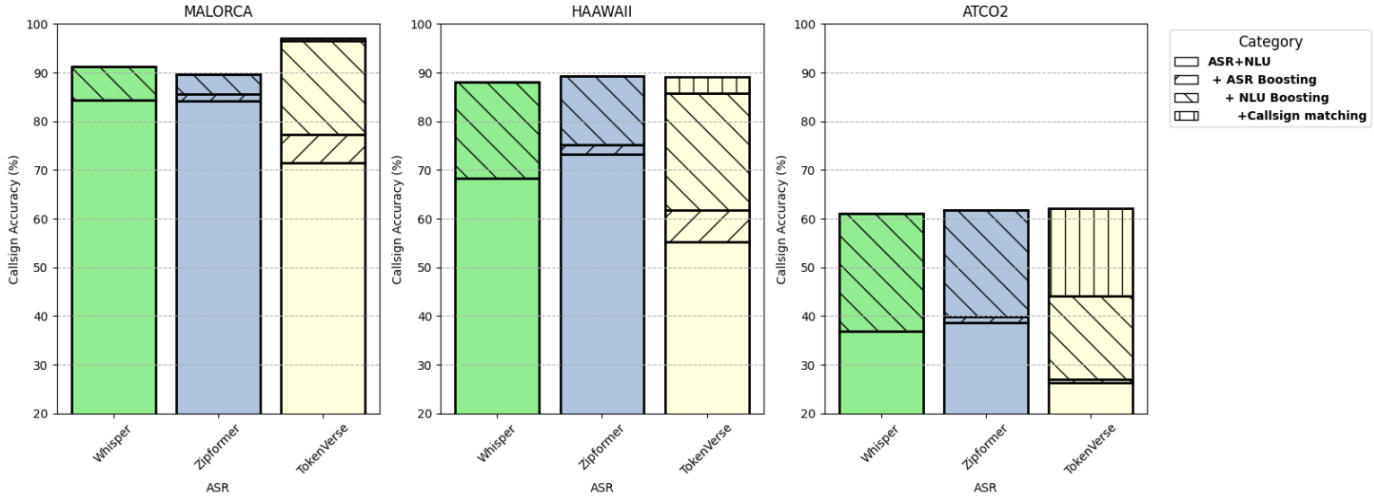


Figure 5. Obtained Callsign detection accuracies for selected Whisper, Zipformer and TokenVerse ASR systems, further trained on 500 h of untranscribed data using SSL.

space of the XLSR-Transducer ASR model, enabling dynamic task activation and allowing training on utterances annotated for only a subset of tasks. TokenVerse++ has been shown to outperform the original TokenVerse model on conversational ASR, and we aim to evaluate its effectiveness in the ATC domain as well.

References

[1] A. Prasad, J. Zuluaga-Gomez, P. Motlicek, S. S. Sarfjoo, N. Iuliia, and K. Vesely, "Speech and natural language processing technologies for pseudo-pilot simulator," in *12th SESAR Innovation Days*, 2022.

[2] D. Guo, Z. Zhang, B. Yang, J. Zhang, and Y. Lin, "Boosting low-resource speech recognition in air traffic communication via pretrained feature aggregation and multi-task learning," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 9, pp. 3714–3718, Sept. 2023.

[3] M. S. Kasttet, A. Lyhyaoui, D. Zbakh, A. Aramja, and A. Kachkari, "Toward effective aircraft call sign detection using fuzzy string-matching between asr and ads-b data," *Aerospace*, vol. 11, no. 1, p. 32, 2023.

[4] H. Helmke *et al.*, "Measuring speech recognition and understanding performance in air traffic control domain beyond word error rates," *Proceedings of the 11th SESAR Innovation Days, Virtual*, pp. 7–9, 2021.

[5] M. Kleinert *et al.*, "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)*, 2021, pp. 1–9.

[6] N. Iuliia, 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 *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2022.

[7] S. Kumar *et al.*, "TokenVerse: Towards Unifying Speech and NLP Tasks via Transducer-based ASR," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 20 988–20 995. [Online]. Available: https://aclanthology.org/2024.emnlp-main.1167/

[8] Y.-C. Chen, S. wen Yang, C.-K. Lee, S. See, and H. yi Lee, "Speech representation learning through self-supervised pretraining and multi-task finetuning," 2021. [Online]. Available: https://arxiv.org/abs/2110.09930

[9] S. Kumar *et al.*, "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*. IEEE, Apr. 2024, pp. 12 592–12 596. [Online]. Available: https://ieeexplore.ieee.org/document/10446130

[10] "All clear phraseology manual," in *Eurocontrol, Brussels, Belgium*, 2011, "[Online; accessed 10-September-2021]".

[11] 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.

[12] 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.

[13] 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.

[14] D. Povey *et al.*, "The Kaldi speech recognition toolkit," in *Proc. of IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[15] 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 *Interspeech*, 2016, pp. 2751–2755.

[16] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[17] 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 *Interspeech*, 2018, pp. 3743–3747.

[18] A. Conneau *et al.*, "Unsupervised cross-lingual representation learning for speech recognition," *arXiv preprint arXiv:2006.13979*, 2020.

[19] A. Baevski *et al.*, "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.

[20] A. Vyas, S. Madikeri, and H. Bourlard, "Lattice-free mmi adaptation of self-supervised pretrained acoustic models," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6219–6223.

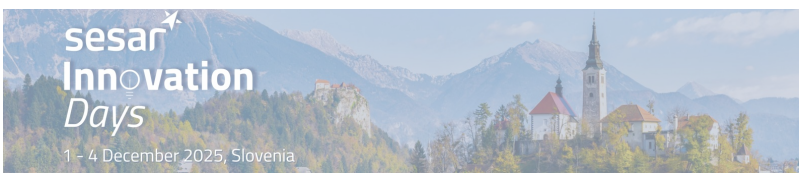
[21] Y. Wang *et al.*, "Espresso: A fast end-to-end neural speech recognition toolkit," in *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 136–143.

[22] M. Ott *et al.*, "fairseq: A fast, extensible toolkit for sequence modeling," *arXiv preprint arXiv:1904.01038*, 2019.

[23] 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.

[24] 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.

[25] D. S. e. Park, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.



[26] J. van Doorn *et al.*, "Whisper-atc: Open models for air traffic control automatic speech recognition with accuracy," in *International Conference on Research in Air Transportation*, 2024, pp. ICRAT–2024.

[27] H. Ma, Z. Peng, M. Shao, J. Li, and J. Liu, "Extending whisper with prompt tuning to target-speaker asr," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 12 516–12 520.

[28] J. J. D. Piaget, A. Prasad, and P. Motlicek, "Improving asr and callsign detection in air traffic control speech using whisper prompting," Idiap, Rue Marconi 19, Martigny, 1920, Idiap-RR Idiap-RR-04-2025, 7 2025, this semester project is done as a collaboration between EPFL and Idiap.

[29] Z. e. Yao, "Zipformer: A faster and better encoder for automatic speech recognition," in *The Twelfth International Conference on Learning Representations*, 2023.

[30] G. C. et.al, "Gigaspeech: An evolving, multi-domain ASR corpus with 10, 000 hours of transcribed audio," in *Interspeech*. ISCA, 2021, pp. 3670–3674.

[31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1810.04805

[32] S. Kumar, S. Madikeri, J. Zuluaga-Gomez, E. Villatoro-Tello, I. Thorbecke, P. Motlicek, M. K. E, and A. Ganapathiraju, "Xlsr-transducer: Streaming asr for self-supervised pretrained models," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, Apr. 2025. [Online]. Available: https://ieeexplore.ieee.org/document/10888110

[33] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing." [Online]. Available: http://arxiv.org/abs/1808.06226

[34] K. Hu *et al.*, "Massively multilingual shallow fusion with large language models." [Online]. Available: http://arxiv.org/abs/2302.08917

[35] A. Andrusenko, A. Laptev, V. Bataev, V. Lavrukhin, and B. Ginsburg, "Fast Context-Biasing for CTC and Transducer ASR models with CTC-based Word Spotter," in *Interspeech 2024*, 2024, pp. 757–761.

[36] M. e. a. Bhattacharjee, "Contextual biasing methods for improving rare word detection in automatic speech recognition," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 12 652–12 656, ISSN: 2379-190X. [Online]. Available: https://ieeexplore.ieee.org/document/10447465

[37] J. Zuluaga-Gomez *et al.*, "Contextual semi-supervised learning: An approach to leverage air-surveillance and untranscribed ATC data in ASR systems." [Online]. Available: http://arxiv.org/abs/2104.03643

[38] A. Carofilis, P. Rangappa, S. Madikeri, S. Kumar, S. Burdisso, J. Prakash, E. Villatoro-Tello, P. Motlicek, B. Sharma, K. Hacioğlu, S. Venkatesan, S. Vyas, and A. Stolcke, "Better semi-supervised learning for multi-domain asr through incremental retraining and data filtering," in *Interspeech 2025*, Aug. 2025, pp. 3618–3622. [Online]. Available: https://www.isca-archive.org/interspeech_2025/carofilis25_interspeech.pdf

[39] P. Rangappa, A. Carofilis, J. Prakash, S. Kumar, S. Burdisso, S. Madikeri, E. Villatoro-Tello, B. Sharma, P. Motlicek, K. Hacioğlu, S. Venkatesan, S. Vyas, and A. Stolcke, "Efficient data selection for domain adaptation of asr using pseudo-labels and multi-stage filtering," in *Proc. Interspeech*, 2025.

[40] S. Kumar, S. Madikeri, E. Villatoro-Tello, S. Burdisso, P. Rangappa, A. Carofilis, P. Motlicek, K. P. D. S, S. Venkatesan, K. Hacioğlu, and A. Stolcke, "Tokenverse++: Towards flexible multitask learning with dynamic task activation," in *2025 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2025.