CONTEXTUAL BIASING METHODS FOR IMPROVING RARE WORD DETECTION IN AUTOMATIC SPEECH RECOGNITION

Mrinmoy Bhattacharjee [1,⋆,†] *Iuliia Nigmatulina* [1,3] *Amrutha Prasad* [1,2] *Pradeep Rangappa* [1]
Srikanth Madikeri [1] *Petr Motlicek* [1,2] *Hartmut Helmke* [4] *Matthias Kleinert* [4]

[1] Idiap Research Institute, Martigny, Switzerland
[2] Brno University of Technology, Czech Republic
[3] University of Zurich, Switzerland
[4] German Aerospace Center (DLR), Institute of Flight Guidance, Braunschweig, Germany

ABSTRACT

In specialized domains like Air Traffic Control (ATC), a notable challenge in porting a deployed Automatic Speech Recognition (ASR) system from one airport to another is the alteration in the set of crucial words that must be accurately detected in the new environment. Typically, such words have limited occurrences in training data, making it impractical to retrain the ASR system. This paper explores innovative word-boosting techniques to improve the detection rate of such rare words in the ASR hypotheses for the ATC domain. Two acoustic models are investigated: a hybrid CNN-TDNNF model trained from scratch and a pre-trained wav2vec2-based XLSR model fine-tuned on a common ATC dataset. The word boosting is done in three ways. First, an out-of-vocabulary word addition method is explored. Second, G-boosting is explored, which amends the language model before building the decoding graph. Third, the boosting is performed on the fly during decoding using lattice re-scoring. The results indicate that the G-boosting method performs best and provides an approximately 30-43% relative improvement in recall of the boosted words. Moreover, a relative improvement of up to 48% is obtained upon combining G-boosting and lattice-rescoring.

Index Terms— Automatic speech recognition, air traffic control, domain adaptation, contextual biasing, rare word recognition

1. INTRODUCTION

Detecting rare words in Automatic Speech Recognition (ASR) is crucial for diverse applications [1], including transcription accuracy, domain-specific terminology, and Air Traffic Control (ATC) communication. Unique waypoint names at airports require accurate recognition. Cost-effective biasing techniques are needed to modify trained ASR systems for rare words. This study analyzes three approaches to enhance the performance of deployed ASR systems to accurately detect a group of words that pose challenges for prediction. These challenges arise because these words are either absent during the training phase or seldom occur.

Recent interest in improving ASR performance on rare words has grown [2]. Sun et al. [3] introduced a novel tree-constrained pointer generator for ASR models to incorporate contextual knowledge through a prefix tree structure efficiently. Their method was shown to improve recognition rates for biasing words consistently. Tong et al. [4] proposed contextual biasing (CB) for personalized speech recognition, enhancing recognition of infrequent words. Qiu et al. [5] focused on confidence estimation, introducing a context-aware model. Sim et al. [6] discussed personalization techniques for mobile devices, considering data privacy. Sainath et al. [7] enhanced contextual biasing by injecting representative text data during training, improving phrase recognition. Fox et al. [8] introduced standardized biasing lists for contextual ASR and an alternate spelling prediction model. Pundak et al. [9] presented the Contextual Listen, Attend, and Spell (CLAS) approach, emphasizing context incorporation. Nigmatulina et al. [10] proposed a two-step approach for improving callsign recognition in ATC involving ASR weight adjustments and NLP-based post-processing that was shown to perform well across various test sets.

Despite these innovations, challenges in the form of resource constraints and model complexity persist. In this context, the present research investigates three recent biasing techniques: adding Out-of-Vocabulary (OOV) words, G-boosting, and lattice-rescoring. The rationale behind exploring these three methods is threefold. First, there is no need for expert knowledge to execute these techniques, and adaptation can be automatized on the end-user side. Second, these approaches seamlessly integrate into the standard ASR system pipeline, as illustrated in Fig. 1, making them suitable to be distributed as black-box utilities for clients who may not be experts in the field. Third, the algorithms are lightweight, and it is relatively easy to balance the degree of word enhancement and overall system performance. Three

⋆Partially supported by DLR internal funding from the DIAL project.

†Corresponding author: *mrinmoy.bhattacharjee@idiap.ch*

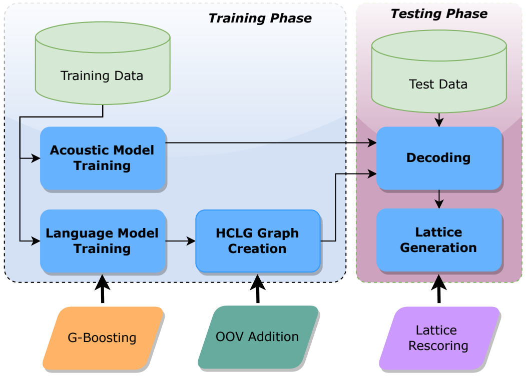

Fig. 1. Block diagram of the word-boosting ASR system.

test sets are used for benchmarking that include proprietary and public data. The pros and cons of each method are analyzed, providing guidelines for choosing the algorithms in various situations.

The paper is organized as follows: Section 2 briefly describes the three biasing methods. Section 3 discusses experimental design and results. Section 4 concludes the paper, suggesting future directions.

2. CONTEXUAL BIASING METHODS

Significant amount of work has been done to test ASR solutions suitable for ATC (e.g. including fully supervised [11], or semi-supervised hybrid approaches [12], or recent large pretrained models [13]). Contextual biasing modifies an ASR system to improve recognition accuracy for specific words or sequences without compromising overall performance [14, 15]. Such methods are particularly crucial when creating a new ASR system is costly or impractical. This study focuses on adapting an existing ASR system for airport domains, where unique important words are rare in training data. Even more significant is that these are artificial terms not commonly found in everyday English dialogue, such as *balad* and *mabod*, used as waypoints for aircraft navigation in Austrian airspace. Typically, these words consist of a combination of two or three vowels and a small number of consonants. Contextual biasing algorithms play a key role in addressing this challenge. Three recent algorithms are discussed in the following paragraphs. It is important to highlight that for all the techniques under discussion in this study, a critical assumption is that the newly introduced words contain no new phones.

2.1. OOV word addition to HCLG by replacing *UNK*

The Kaldi toolkit [16] constructs the decoding graph as a Finite State Transducer (FST) using the composition of $HCLG$. Here, G represents the language model as an acceptor, L serves as the lexicon mapping phones to words, C transforms context-dependent phones to context-independent ones, and H contains HMM definitions mapping transition ids to context-dependent phones. Adding OOV words to the HCLG graph through $[unk]$ arcs is proposed in [17] and consists of two core steps. First, the original $HCLG$ graph is composed using the large dictionary used from training data. Secondly, $L.fst$ and HCL are created with the new (OOV) words. Subsequently, these words are introduced into the previously created $HCLG$ graph using the pronunciations from the new HCL and the $[unk]$ arcs. An important condition of this method is that the LM is trained so that $[unk]$ can only appear at the end of an n-gram (can be achieved when trained with the $pocolm$ with the $limit-unk-history$ option). This step allows the insertion of an OOV HCL just once, pointing arcs matching $[unk]$ to it. The original $[unk]$ weight can also be adjusted for OOVs.

2.2. G boosting

This method assumes that the words to be boosted are known as a priori and present in the dictionary. In this approach [18], target words and/or word sequences are boosted by modifying the n-gram language model ($G.fst$) built from the training data before the decoding step. While iterating over the arcs in the baseline $G.fst$, weights of the existing arcs that match the target words are updated by a constant discount $-\log p$. The corresponding arcs are created with small weights if they do not exist. The updated $G.fst$ with more prominent weights for the target words is then composed with the rest of the decoding graph HCL for the decoding.

Let $\mathcal{B}_i$ be a word in a list of N rare words that need to be boosted in $G.fst$, where $i=1,\ldots,N$. Also, let $\mathcal{A}^j_{\mathcal{B}_i}$ indicate the j^{th} arc in $G.fst$ that has the input label as $\mathcal{B}_i$, output label as $\mathcal{B}_i$, and the arc weight as $W\left(\mathcal{A}^j_{\mathcal{B}_i}\right)_{old}$. The boosting operation can be represented by Eq. 1.

$$W\left(\mathcal{A}^j_{\mathcal{B}_i}\right)_{new} = W\left(\mathcal{A}^j_{\mathcal{B}_i}\right)_{old} - \log(p) \tag{1}$$

The boosting operation is performed for all arcs representing the list of words to be boosted. When boosting a sequence of words, a new arc is added with a preset weight in the $G.fst$ if the arc for a particular word does not exist in the given context. The operation performed using eqn. 1 enhances the probability of the word (or sequence of words) being selected in the top hypothesis while decoding. It is to be noted that $G.fst$ weights are negative log probabilities, where the weight (positive number) is inversely proportional to the probability. Hence, a lower probability corresponds to higher weight.

2.3. Lattice rescoring

In lattice rescoring, the weights of target words and (or) word sequences are updated directly in the decoding lattices [19]. A bias FST is first created, which includes all target words and word sequences with a discount factor on their arcs. Then, the rescoring is typically done as the composition of a lattice with the bias FST, which leads to the target weight adjustment directly before the final prediction.

3. EXPERIMENTAL RESULTS

This section presents the performance of three word boosting algorithms, including overall Word Error Rate (WER) and rare word Precision, Recall, and F1-score. The next subsection provides details on the ASR models used in this study.

3.1. ASR model

This work strictly considers only a monolingual English ASR system. To train the initial acoustic model and conduct decoding and rescoring experiments, we utilized the Kaldi framework [16]. Two types of acoustic models are analyzed in this work. First, a smaller hybrid-based CNN-TDNNF model trained from scratch on ATC labeled data [20]. This model was trained using the best-known Kaldi recipe employing Lattice-Free Maximum Mutual Information (LF-MMI) architecture with effective GPU parallelization (natural gradient descent) applied during training, using MFCC and i-vector features. The training methodology employed LF-MMI loss [21], encompassing a 3-fold speed perturbation and one-third frame sub-sampling. Second, the XLSR model [22] pre-trained with a dataset as large as 56k hours of speech data is fine-tuned using the same data as in CNN-TDNNF model applying the approach described in [23]. The authors in [23] propose to use the LF-MMI criterion (similar to hybrid-based ASR) for the supervised adaptation of the self-supervised pre-trained XLSR model [22]. This approach has been shown that it can outperform [20] the models trained with only the supervised data. A 3-gram language model trained on the same data as the acoustic model and supplemented with textual data from additional public resources such as airline names, airport information, ICAO alphabet, and European waypoints.

3.2. Datasets

For training (or fine-tuning) the acoustic models, 195 hours of labeled ATC data have been used [20]. The training data is generated from multiple ATC databases as a result of applying speed perturbation. Three evaluation datasets are used in this work. The first two evaluation test sets are proprietary data from the funding agency to evaluate the developed systems. The first of these test sets will be subsequently referred to as Proprietary Test Set 1 (PTS-1), while the second one will be called Proprietary Test Set 2 (PTS-2). The data were collected during proof-of-concept exercises during real communication between air traffic controllers and pilots. The recording conditions were relatively clean despite the varied English accents of the speakers. Moreover, the exercises were comprised of speech utterances which contained a set of words not seen (or rarely seen) during training. PTS-1 consists of a total of 128 utterances, while PTS-2 consists of 77 utterances. A set of 11 unique waypoints were identified in the test set that were poorly recognized by the baseline ASR system and needed to be boosted. These words appeared for a total of 21 times in PTS-1 and 75 times in PTS-2. The precision, recall, and f1-score reported in subsection 3.3 are based on these statistics.

Table 1. Performance on ***PTS-1*** using both CNN-TDNNF and XLS-R based acoustic models. The best results are highlighted in **boldface**. Here, OA := OOV Addition, GB := G-boosting, LR := Lattice Rescoring.

			CNN-TDNNF				**XLSR**			
			WER	**Rare word**			**WER**	**Rare word**		
OA	**GB**	**LR**		**Prec**	**Rec**	**F1**		**Prec**	**Rec**	**F1**
-	-	-	25.47	**1.0**	0.05	0.09	16.22	**1.0**	0.29	0.44
✓	-	-	25.57	0.0	0.0	0.0	17.11	**1.0**	0.14	0.25
-	✓	-	24.48	0.28	0.52	0.36	**16.13**	0.54	**0.95**	0.69
-	-	✓	24.78	**1.0**	0.24	**0.38**	16.22	**1.0**	0.57	0.73
✓	-	✓	**23.80**	**1.0**	0.05	0.09	16.72	**1.0**	0.24	0.38
-	✓	✓	24.48	0.26	**0.57**	0.35	**16.13**	0.67	**0.95**	**0.78**

The third evaluation test set used in this work corresponds to the 1.1 hours of open-source transcribed annotations from the ATCO2 corpus. The ATCO2 project [1] was designed to create a distinctive platform that can gather, arrange, and prepare air-traffic control voice communication data from airspace. The ATCO2 corpus was built to develop and evaluate ASR and NLP technologies for English ATC communications. The dataset comprises English voice data from several airports worldwide (e.g., Brno, Prague, Bratislava, Sion, Zurich, Bern, and Sydney). This test set can be accessed for free [2]. For evaluating the boosting performance on the ATCO2 data, 12 poorly recognized waypoints were selected that appeared 191 times in this test set.

3.3. Results

The performance of the methods on OOV addition (see 2.1), G boosting (see 2.2), and lattice rescoring (see 2.3) are reported for the three test sets mentioned above. For PTS-1 data (see Table 1), with CNN-TDNNF acoustic model lattice rescoring achieves the highest rare word F1-score (0.38), while G boosting + lattice rescoring has the best rare word recall (0.57). With XLSR, G-boosting leads in rare word recall (0.95), while G boosting + lattice rescoring excels in F1-score (0.78). For the PTS-2 data (see Table 2), G-boosting achieves the highest rare word recall (0.45) and F1-score (0.62) with CNN-TDNNF acoustic model. With XLSR, G boosting + lattice rescoring excels in rare word recall (0.77) and F1-score (0.87), but G-boosting alone has the best overall WER (7.81). For the ATCO2 test set (see Table 3), combining G-boosting and lattice rescoring yields the best rare word detection (WER: 26.15, recall: 0.66, F1-score: 0.73) with CNN-TDNNF. With XLSR, G-boosting + lattice rescoring achieves the highest rare word detection recall (0.88), while G-boosting alone has the best F1-score (0.88).

3.4. Effect of tuning the discount factor

Figure 2 illustrates the impact of adjusting the discount factor for G-boosting (mentioned in subsection 2.2). The x-axis

[1] https://catalog.elra.info/en-us/repository/browse/ELRA-S0484/

[2] https://www.atco2.org/data (accessed on 12 May 2023)

Table 2. Performance on ***PTS-2*** using both CNN-TDNNF and XLSR based acoustic models. The best results are highlighted in **boldface**. Here, OA := OOV Addition, GB := G-boosting, LR := Lattice Rescoring.

			CNN-TDNNF				XLSR			
			WER	Rare word			WER	Rare word		
OA	**GB**	**LR**		**Prec**	**Rec**	**F1**		**Prec**	**Rec**	**F1**
-	-	-	15.95	**1.0**	0.12	0.21	7.97	**1.0**	0.52	0.68
✓	-	-	15.95	**1.0**	0.11	0.19	11.79	**1.0**	0.35	0.51
-	✓	-	11.13	0.97	**0.45**	**0.62**	**7.81**	0.98	0.75	0.85
-	-	✓	11.63	**1.0**	0.39	0.56	8.14	**1.0**	0.68	0.81
✓	-	✓	15.28	**1.0**	0.19	0.31	10.47	**1.0**	0.51	0.67
-	✓	✓	**10.96**	0.97	**0.45**	**0.62**	7.97	0.98	**0.77**	**0.87**

Table 3. Performance on ***ATCO2 test set*** using both CNN-TDNNF and XLSR based acoustic models. The best results are highlighted in **boldface**. Here, OA := OOV Addition, GB := G-boosting, LR := Lattice Rescoring.

			CNN-TDNNF				XLSR			
			WER	Rare word			WER	Rare word		
OA	**GB**	**LR**		**Prec**	**Rec**	**F1**		**Prec**	**Rec**	**F1**
-	-	-	26.84	**0.91**	0.39	0.55	17.53	0.94	0.42	0.58
✓	-	-	26.84	**0.91**	0.39	0.55	18.53	**0.96**	0.38	0.54
-	✓	-	26.20	**0.91**	0.61	**0.73**	**16.37**	0.92	0.85	**0.88**
-	-	✓	26.84	0.86	0.42	0.57	18.11	0.89	0.52	0.66
✓	-	✓	26.84	0.86	0.42	0.57	17.61	0.89	0.51	0.65
-	✓	✓	**26.15**	0.8	**0.66**	**0.73**	17.10	0.83	**0.88**	0.85

represents the discount factor values (p), and the y-axis shows scaled overall WER and rare word recall. Low p values do not enhance rare word recall. However, as p gradually exceeds 1.0, recall improves while WER decreases. Beyond $p \approx 1.3$, WER drops sharply due to boosted words being falsely predicted for many other words in the ground truth. These results suggest a limit to boosting certain words without adversely affecting the overall WER for the test set.

3.5. Discussions

The initial insight gained from Tables 1, 2, and 3 indicates that a superior acoustic model tends to improve results when used with boosting algorithms. Nevertheless, these boosting techniques exhibit similar performance trends across acoustic models. Another key finding is that the OOV word addition approach performs worse than the other two methods. G-boosting emerges as the top-performing approach, with lattice-rescoring following closely. Combining G-boosting and lattice-rescoring sometimes improves upon their separate performances. Lastly, the choice of the discount factor in G-boosting significantly influences the recognition of rare words, with very low values having little impact and high values causing over-prediction of boosted words.

The three methods studied in this work have different requirements and performance. OOV addition is flexible but has poor performance. It can be used even in cases where the set of important words is not part of the ASR dictionary. G-boosting works best but requires words to be in the dictionary. Lattice-rescoring is easy to implement and second-best in performance but requires that the words be present both in the dictionary and n-best decoded lattices. The best method to use depends on the specific scenario. The ease of implementation should also be a factor to be considered.

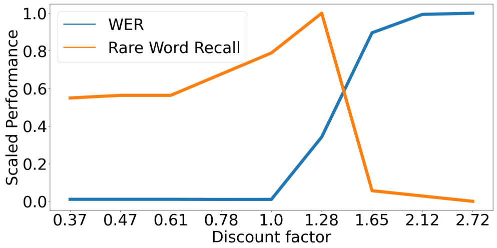

Fig. 2. Effect of discount factor on effective word boosting.

4. CONCLUSION

This research investigates word-boosting algorithms aimed at enhancing the detection of rare words within the Air Traffic Control (ATC) domain, where the demand for such methods is critically significant. Through a comprehensive analysis, we examine three recently proposed techniques, namely OOV addition, G-boosting, and lattice-rescoring. Our study uses two distinct types of acoustic models: a smaller CNN-TDNNF model trained from scratch and a larger pre-trained XLSR model fine-tuned on the same dataset. While boosting generally exhibits superior performance with the larger acoustic model, the observed performance trends remain consistent across both model types. The results of this study indicate that G-boosting emerges as the most effective approach among the trio, with lattice-rescoring following closely behind. The overall best performance is provided by the combination of G-boosting and lattice-rescoring, providing a relative improvement of up to 50% in the F1-score for the ATCO2 test set. In future research endeavors, it may be worthwhile to explore the integration of OOV addition and G-boosting methods, thereby facilitating the adaptation of existing ASR systems for detecting out-of-vocabulary words and enhancing their recognition performance while incurring minimal associated costs.

5. REFERENCES

[1] C. H. Yang, L. Liu, A. Gandhe, Y. Gu, A. Raju, D. Filimonov, and I. Bulyko, "Multi-task language modeling for improving speech recognition of rare words," in *Proc. Workshop on Autom. Speech Recognit. and Understanding*, 2021, pp. 1087–1093.

[2] T. Munkhdalai, Z. Wu, G. Pundak, K. C. Sim, J. Li, P. Rondon, and T. N. Sainath, "NAM+: Towards Scalable End-to-End Contextual Biasing for Adaptive ASR," in *Proc. Spoken Lang. Tech. Workshop*, 2023, pp. 190–196.

[3] G. Sun, C. Zhang, and P. C. Woodland, "End-to-End Spoken Language Understanding with Tree-Constrained Pointer Generator," in *Proc. Int. Conf. on Acoust., Speech and Signal Process.*, 2023, pp. 1–5.

[4] S. Tong, P. Harding, and S. Wiesler, "Slot-Triggered Contextual Biasing For Personalized Speech Recognition Using Neural Transducers," in *Proc. Int. Conf. on Acoust., Speech and Signal Process.*, 2023, pp. 1–5.

[5] D. Qiu, T. Munkhdalai, Y. He, and K. C. Sim, "Context-Aware Neural Confidence Estimation for Rare Word Speech Recognition," in *Proc. Spoken Lang. Tech. Workshop*, 2023, pp. 31–37.

[6] K. C. Sim, F. Beaufays, A. Benard, D. Guliani, A. Kabel, N. Khare, T. Lucassen, P. Zadrazil, H. Zhang, L. Johnson, G. Motta, and L. Zhou, "Personalization of End-to-End Speech Recognition on Mobile Devices for Named Entities," in *Proc. Autom. Speech Recog. and Understanding Workshop*, 2019, pp. 23–30.

[7] T. N. Sainath, R. Prabhavalkar, D. Caseiro, P. Rondon, and C. Allauzen, "Improving Contextual Biasing with Text Injection," in *Proc. Int. Conf. on Acoust., Speech and Signal Process.*, 2023, pp. 1–5.

[8] J. Fox and N. Delworth, "Improving Contextual Recognition of Rare Words with an Alternate Spelling Prediction Model," in *Proc. Interspeech*, 2022, pp. 3914–3918.

[9] G. Pundak, T. N. Sainath, R. Prabhavalkar, A. Kannan, and D. Zhao, "Deep Context: End-to-end Contextual Speech Recognition," in *Proc. Spoken Lang. Tech. Workshop*, 2018, pp. 418–425.

[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 *Proc. Int. Conf. on Acoust., Speech and Signal Process.*, 2022, pp. 6282–6286.

[11] J. Zuluaga-Gomez, P. Motlicek, Q. Zhan, R. Braun, and K. Vesely, "Automatic Speech Recognition Benchmark for Air-Traffic Communications," in *Proc. Interspeech 2020*, Oct 2020, pp. 2297–2301.

[12] M. Kleinert, H. Helmke, G. Siol, H. Ehr, A. Cerna, C. Kern, D. Klakow, P. Motlicek, Y. Oualil, M. Singh, and A. Srinivasamurthy, "Semi-supervised Adaptation of Assistant Based Speech Recognition Models for different Approach Areas," in *Proc. 37th Digital Avionics Systems Conf. (DASC)*, 2018, pp. 1–10.

[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 *Proc. Spoken Lang. Tech. Workshop*, Jan 2023, vol. 1 of *1*.

[14] D. Zhao, T. N. Sainath, D. Rybach, P. Rondon, D. Bhatia, B. Li, and R. Pang, "Shallow-Fusion End-to-End Contextual Biasing," in *Proc. Interspeech*, 2019, pp. 1418–1422.

[15] M. Bhattacharjee, P. Motlicek, I. Nigmatulina, H. Helmke, O. Ohneiser, M. Kleinert, and H. Ehr, "Customization of Automatic Speech Recognition Engines for Rare Word Detection Without Costly Model Re-Training," in *Proc. 13th SESAR Innovation Days*, Nov 2023.

[16] 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 *Proc. Workshop on Autom. Speech Recognit. and Understanding*, 2011.

[17] R. A. Braun, S. Madikeri, and P. Motlicek, "A comparison of methods for oov-word recognition on a new public dataset," in *Proc. Int. Conf. on Acoust., Speech and Signal Process.*, 2021, pp. 5979–5983.

[18] I. Nigmatulina, R. Braun, J. Zuluaga-Gomez, and P. Motlicek, "Improving callsign recognition with air-surveillance data in air-traffic communication," in *arXiv preprint arXiv:2108.12156*, 2021, pp. 1–5.

[19] M. Kocour, K. Veselỳ, A. Blatt, J. Zuluaga-Gomez, I. Szöke, J. Cernocky, D. Klakow, and P. Motlicek, "Boosting of Contextual Information in ASR for Air-Traffic Call-Sign Recognition," in *Proc. Interspeech*, 2021, pp. 3301–3305.

[20] J. Zuluaga-Gomez, A. Prasad, I. Nigmatulina, P. Motlicek, and M. Kleinert, "A Virtual Simulation-Pilot Agent for Training of Air Traffic Controllers," *MDPI Aerospace*, vol. 10, no. 5, 2023.

[21] 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, pp. 2751–2755.

[22] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," *arXiv preprint arXiv:2006.13979*, 2020.

[23] A. Vyas, S. Madikeri, and H. Bourlard, "Lattice-Free Mmi Adaptation of Self-Supervised Pretrained Acoustic Models," in *Proc. Int. Conf. on Acoust., Speech and Signal Process.*, 2021, pp. 6219–6223.