Customization of Automatic Speech Recognition Engines for Rare Word Detection Without Costly Model Re-Training

Mrinmoy Bhattacharjee [1], Petr Motlicek [1,2],
Iuliia Nigmatulina [1,3]
[1] Idiap Research Institute, Martigny, Switzerland
[2] Brno University of Technology, Czech Republic
[3] University of Zurich, Switzerland
firstname.lastname@idiap.ch

Hartmut Helmke, Oliver Ohneiser,
Matthias Kleinert, Heiko Ehr
Institute of Flight Guidance,
German Aerospace Center (DLR)
Braunschweig, Germany
Firstname.Lastname@dlr.de

***Abstract*—Thanks to Alexa, Siri or Google Assistant automatic speech recognition (ASR) has changed our daily life during the last decade. Prototypic applications in the air traffic management (ATM) domain are available. Recently pre-filling radar label entries by ASR support has reached the technology readiness level before industrialization (TRL6). However, seldom spoken and airspace related words relevant in the ATM context remain a challenge for sophisticated applications. Open-source ASR toolkits or large pre-trained models for experts – allowing to tailor ASR to new domains – can be exploited with a typical constraint on availability of certain amount of domain specific training data, i.e., typically transcribed speech for adapting acoustic and/or language models. In general, it is sufficient for a "universal" ASR engine to reliably recognize a few hundred words that form the vocabulary of the voice communications between air traffic controllers and pilots. However, for each airport some hundred dependent words that are seldom spoken need to be integrated. These challenging word entities comprise special airline designators and waypoint names like "dexon" or "burok", which only appear in a specific region. When used, they are highly informative and thus require high recognition accuracies. Allowing plug and play customization with a minimum expert manipulation assumes that no additional training is required, i.e., fine-tuning the universal ASR. This paper presents an innovative approach to automatically integrate new specific word entities to the universal ASR system. The recognition rate of these region-specific word entities with respect to the universal ASR increases by a factor of 6.**

Keywords—Speech Recognition; Model Adaptation; Integration of prior knowledge; Customization of model, Rare-word integration.

I. INTRODUCTION

A. Problem and Challenges

DLR and MITRE have analyzed millions of word entities from voice communication between air traffic controllers (ATCos) and pilots. The most frequently used words are typically the ten digits, viz., one, two, three, … nine, zero, which cover roughly 40% of the spoken words [1]. 550 words cover 95% of the spoken words in the US data. Nevertheless, the dictionary size that is used in ATCo–pilot voice communications is much larger (order of tens of thousands different word entities). Nearly 10,000 airline designators like *speed bird*, *egyptian bird* or *ocean* exist. The number of waypoints used across airports and airspaces, fixes or navigation aid names all over the world like *DEXON, MOBSA, or DOMUX* is even bigger than 10,000. To make it even more challenging, parts of this word list are typically updated every month, i.e., new word entities appear so that speech recognition and downstream understanding engines deteriorate over time. This is a serious lifecycle maintenance issue. It is a particularly large challenge for applications that need to be scaled up to cover multiple air traffic control (ATC) sectors and facilities.

One of the first applications of automatic speech recognition (ASR) in air traffic management was the support or replacement of simulation pilots by ASR [2]. The first applications were used by air navigation service providers (ANSPs), which simulated one or two airspaces. The set of waypoints was rather static. When changing to a new airspace (e.g., simulation of Frankfurt approach instead of Heathrow approach), the list of waypoints completely changes. The DIAL project [3], conducted by six DLR institutes, is supported by a universal ASR engine of Idiap that was originally developed for the Vienna approach area. DIAL, however, considers the Celle sector in upper airspace, usually controlled by Maastricht upper airspace center (UAC), as relevant airspace [4]. For the original Vienna approach use-case, 260 waypoints such as *ABIRI* or *BALAD* have been modelled. Later on, 565 different waypoints were required to be well recognized by ASR for the new use-case in DIAL, e.g., *KOSEK*, *WYK*, or *DOR* (*DOR* is pronounced as "*wickede*" being a village in western part of Germany and *WYK* is pronounced as "*wipper*"). Contrary to this, "*kosek*" for Celle sector or "*abiri*" for Vienna approach are artificial 5-letter words composed of vowels and consonants. All these words are typically not found in a dictionary of the universal ASR engine (since these words were not found in training material). Sometimes the pronunciation of these words can easily be derived from the spelling, while at other times this is very difficult. Sometimes the pronunciation is even ATCo dependent, especially for towns containing letters not part of the 26 letters used

in normal English e.g. “Osnabrück” in German or “Liège” in French.

The DIAL project aims to develop a digital air traffic controller assistant. The assistant is supposed to manage less challenging aircraft without the intervention of the ATCo. This way, the ATCo can concentrate on more challenging aircraft. Furthermore, DIAL supports simulation pilots by a method called automatic speech recognition and understanding (ASRU). Integration of ASRU helps to reduce the number of required simulation pilots per exercise. The past validation trials with a universal ASRU solution developed for Vienna approach in DLR’s lab environment were based on 120,000 word entities resulting in an overall word error rate (WER) of the ASR engine of 3.1% [5]. However, when applying this ASRU solution from past projects during the first trials of DIAL, WER has increased to a high WER of 13.9% (see TABLE VI. in subsection V.B). Such a degradation in performance is specifically due to the presence of waypoint names that were not seen during the ASR training. The statistical system that models the language structure relies on the surrounding context while generating the transcript for a specific speech segment. This implies that the poor detection of waypoints also degrades the detection of words around these waypoints. Conversely, retraining an ASR system for a new airspace with a different set of waypoint names is challenging. This paper proposes an easy-to-use solution for tackling the aforementioned problem.

B. *Paper Structure*

This paper presents an overview on previous work to tackle the Out of Vocabulary (OOV) problem in Section II and briefly discusses the current state-of-the-art in ASR for ATC. Section III describes the experimental setup to test the recognition of OOV words after customization. Section IV describes different solutions, which have been implemented in the context of the DIAL project. Section V presents the results with respect to word level and semantic level performance. Section VI discusses the current status and suggestions, how others can benefit from the already achieved results, before section VII presents the conclusions.

II. Current State-of-the-art ASR Solution for ATC

A. *Automatic Speech Recognition*

ASR, also often referred to as speech-to-text system, automatically converts the input speech to a textual form, i.e., a sequence of words. In the case of ATC communication, we mean the voice conversation captured by microphones on the side of ATCos or pilots with input speech. The most advanced ASR technology developed in recent past for ASR for ATM applications comes from HAAWAII project [6] (HAAWAII = Highly Advanced Air Traffic Controller Workstation with Artificial Intelligence Integration). The project, although focusing on innovative ASR approaches in ATM, also required a certain level of maturity so that the developed ASR solution can provide needed recognition accuracies, which can then be used for subsequent downstream applications, e.g., callsign highlighting [7], pre-filling radar label entries [5], or readback error detection [8]. Furthermore, we have decided to re-use a hybrid ASR solution that was trained on relatively large manually transcribed speech data available from HAAWAII project as well as from other past projects. As shown in Fig. 1 the ASR engine consists of a combination of independently trained acoustic models (AM) and language models (LM). In this work, the acoustic model is trained as a classical deep neural network instead of using new end-to-end architectures [9]. A hybrid-based ASR system employs separate AM and LM. The AM is trained with a set of speech recordings with a corresponding text transcript, while LM is trained on text only (e.g., the text corresponding to the speech recordings available for training AM is typically used). However, in general much larger textual resources are available than speech data. The AM represents the relationship between a speech signal and phonemes, or other linguistic units, that make up the speech. The LM is usually represented by a probability distribution over sequences of words. The LM in the form of a Finite State Transducer (FST) provides context to distinguish between words and phrases that sound similar. Using the knowledge of AM and LM, a decoding graph is usually built as a Weighted Finite State Transducer (WFST) [10],[11] using the open-source library called OpenFST [12]. The WFST graph generates text output given an observation sequence as shown in Fig. 1. A decoder module uses the decoding graph to predict the best probable transcript corresponding to an input speech signal.

In order to train the hybrid-ASR, additional knowledge is required for its development. In case of monolingual ASR system (English in this case), the minimum knowledge relevant for ATM is a set of phonemes and an input dictionary. Both play an important role in further model customizations. The ATC lexicon consists of words that usually do not appear in English

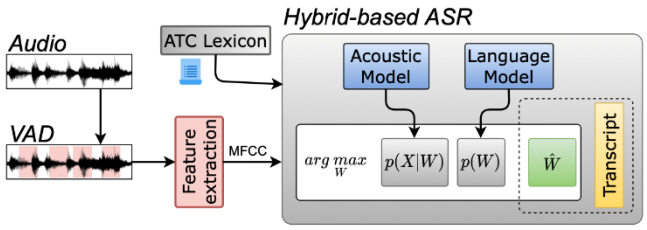

Figure 1. Hybrid-based ASR, where acoustic model combines hidden Markov models and modern deep neural network architectures. Language model is then used during the decoding phase.

conversations, but are specific to ATC. On the one hand, there are frequently used and static terms such as “QNH” and “wilco”. On the other hand, there are seldom used and dynamically changing words (such as artificial waypoint names). Such words may have never been seen during training the ASR model, but might occur in the field when the ASR system is deployed. This work proposes solutions to easily customize the running ASR system to better detect such words in the field.

B. *Acoustic Modelling (AM)*

The acoustic model in hybrid ASR is built around Hidden Markov Models (HMM) combined with Deep Neural Networks

(DNN) [13]. DNNs are an effective module allowing to estimate the posterior probability of a given set of phonemes, or more specifically context-dependent phonemes called triphones. These posterior probabilities can be seen as pseudo-likelihoods or "scaled likelihoods", which can be interfaced with HMM modules. HMMs provide a structure for mapping a temporal sequence of acoustic features extracted from the input speech, e.g., mel frequency cepstral coefficients (MFCC), to a sequence of states [14]. In recent years, end-to-end models are becoming popular as they can be trained as non-autoregressive systems that can model the long future context during training. Nevertheless, hybrid ASR systems still remain one of the best approaches for building production engines, allowing to reach high recognition accuracies. HMM-DNNs based ASR is the state-of-the-art systems for ASR in ATC domain. It was also used in HAAWAII project [8] as well as DIAL project. The acoustic model in those projects was trained in a supervised mode, i.e., manually transcribed data is needed for the target ATC domain.

Some of the recent works have shown further improvements with semi-supervised model training. Such a type of training relies on exploiting automatically labeled speech, i.e., using some universal ASR engine, or an ASR engine developed using a small amount of manually transcribed speech recordings. More details on leveraging non-transcribed ATC speech data by semi-supervised learning can be found in [15], [16]. An advantage of semi-supervised learning is that a large set of unlabeled speech data is easily available and can be employed for training. Large here means 10 or 100 times more than in case of manually transcribed data.

One of the sources of reliable large-scale collection of ATC speech data from different airports worldwide is available from the ATCO2 project [17]. Additionally, innovative research targeted to improve word recognition belonging to the callsign is possible by integrating surveillance data into the pipeline [16], [18].

C. Language Modeling (LM), Dictionary

As part of hybrid-based ASR, LM still plays a crucial role [19]. The main advantage of deploying LM is its large power to bring the generic ASR technology to the target, i.e., the ATC domain. Standard hybrid-based ASR approaches still rely on word-based dictionary as is the case of ASR solution developed for HAAWAII and DIAL projects. The LM can directly be trained on word transcripts collected from the domain. Therefore, word-based LM is used across the ATC scenarios in this paper. In our case, an n-gram LM is deployed with n equal to 3. One of the disadvantages of the word-based approach is that the ASR engine cannot directly recognize words not seen during the training. A set of a-priori known words is required and must be given to the ASR system as dictionary. Its typical size is thousands of words for ATC domain, and hundreds of thousands of words for generic conversational English tasks. This issue is nevertheless addressed by a customization step described later (see section IV).

D. Recognition Process

The process of recognition, i.e., generating the recognition output from the input speech is briefly described here. Trained acoustic and language models are combined together in case of hybrid-ASR solution applied for HAAWAII and DIAL projects. These models are combined using the concept of FST leveraged through the Kaldi framework [20] one of the main streams applied by researchers and companies for ASR.

Trained acoustic model, i.e., specifically the HMM topology, and language models are combined through the concept of FST together with a dictionary and the final graph is used through the process called "decoding". During decoding, the input speech is first used to extract speech features (above mentioned MFCCs), which are then inferred through the DNN architecture. The output set of phonemes represented by a set of posterior probabilities is passed to the decoder to map phoneme sequence to the most likely word sequence using the FST graph. The output then can be seen as a set of word recognition hypotheses, i.e., word sequences represented by data structures called lattices. The lattices carry not only information about word sequences, but also information about confidence for each word. Decoding can be run in an offline mode, i.e., after detecting the end-point in utterance the speech is decoded and the recognized word sequence is returned. Whereas, in online decoding mode, partial word recognition is available in real-time during the process of decoding with a minimum latency of 200-300ms.

III. Experimental Setup

A. Data

The data for training the AM for DIAL project is partially leveraged from the past works. This dataset stems from an exploratory research initiative with the goal of investigating and creating a dependable and flexible system for automatically transcribing voice commands provided by ATCos and pilots alike. The dialogues between ATCos and pilots were sourced from two air navigation service providers (ANSPs): (i) NATS for London approach and (ii) ISAVIA for Icelandic en-route. For training the acoustic models, in total 195 hours of labeled ATC data have been used [9]. The training data is augmented with other internal ATC databases and also using speed perturbation during training. The corresponding reference text has been used to train the baseline language model. The language model is basically a 3-gram model trained using the SRILM toolkit in Kaldi.

The testing data for evaluating the ASR system was collected through proof-of-concept exercises involving ATC utterances from ATCos to pilots. Despite the diverse English accents of the speakers, the recording conditions were generally clean. Furthermore, the exercises included spoken words that were not encountered often or at all during training. Additionally, utterances that especially contain rare words, e.g., waypoints, in the correct context were recorded by different speakers, e.g., "*air france two six alfa proceed hamm*". The test set consists of approximately 52 minutes of audio data with a total of 673 test utterances, comprising a total of 1157 commands (like CON-

TACT, CONTACT_FREQUENCY, DIRECT_TO, etc.). According to the European annotation ontology [22], a "*command*" is a high-level concept that represents an instruction. The number of waypoints considered for boosting in this work are discussed in the next subsection.

B. *Waypoints*

The challenge addressed by this paper is related to the contextual mismatch of test data with respect to training data, specifically for the incorporation of waypoint (word) entities. Waypoints are specialized terms in the field of ATC, corresponding to coordinates. ATCos and pilots use these terms to follow or adjust flight paths. Typically, these waypoint terms are specific to particular areas. For example, if we focus on a specific domain like en-route navigation for Germany, the waypoints used are tailored to that region. However, if we were to apply this system to en-route navigation in Austria, the waypoints would likely change. Consequently, the system might not readily recognize them. In this context, our current work empowers customers to easily incorporate these new waypoints into the system, even without specialized expertise.

Waypoints are names given to a latitude-longitude pair representing a geographic location. A waypoint name such as "WYK" (pronounced as "wipper") is represented by an abbreviation consisting of a sequence of letters or numbers like "DL455". These waypoint names may appear in ATCo-pilot communication as "wipper", "whisky yankee kilo" or "delta lima four five five". When the waypoint is referred to by pronouncing its sequence of letters, the ASR system can easily detect it, as the ICAO phonetic alphabet is commonly encountered as part of the English dictionary during training. However, challenges arise when ATCos or pilots use the artificially created waypoint name like "wipper", which are either newly introduced or infrequently encountered during model training.

TABLE I. STATISTICS RELATED TO THE WORD BOOSTING TASK

Test set size	673 utterances / 52 minutes audio
Number of different words in Dictionary	30,821
Number of unique Waypoints	565
Number of Waypoints in test set	83
Total occurrences of Waypoints	443

TABLE I. lists important statistics about the test data and the waypoints selected for boosting in the present case. The test dataset comprises 673 spoken statements, which correspond to 52 minutes of audio data. The DIAL data base consists of a list of 565 rare words that were required to be detected correctly by the ASR system. From this list of rare words, 84 unique word entities were present in the test set collected for evaluating the ASR system and these occurred for a total of 443 times. However, a fraction of the rare words was never seen during training the LM and can be referred to as OOV words. Therefore, in order to correctly evaluate the performance of the proposed word-boosting methods in improving the detection of the 84 rare words in the test set, the set of OOV words were manually added (see section VI) to the LM and assigned the minimum possible probability. This step is important since the proposed customization methods assume that the words to be boosted are in the dictionary. Nevertheless, this step is easily performed with the tools developed in this work. Once all the 84 words are known to the ASR engine, we use the boosting methods presented in this work to improve the weights of all the 84 words. With this background on the training and testing data, we will present the baseline performance in the following subsection.

C. *Baseline ASR Performance*

In this subsection, the performance of the baseline ASR system is described. Originally, an ASR system trained for different airport/airspace scenarios was developed. However, when the same system was evaluated on the test data, it was observed that the important waypoints and other airport dependent names were not properly recognized. The required customization of this ASR system forms the crux of this work. However, before presenting the results obtained upon the model customization, it is necessary to discuss the results for the unmodified system that will serve as the baseline for this work.

We use four metrics to report the results. The WER measures the percentage of erroneous insertions, substitutions and deletions caused by the ASR system with respect to the total number of reference words in the test data. This metric covers all words and not only the performance on the rare words. A lower WER is preferred. Next, we report the precision, recall and F1-score of detecting the rare words in the test data. For validating or falsifying the hypotheses of the last subsection we use the metrics introduced in [21] and detailed by Chen et al. [22], resulting in a simple scheme for measuring performance on semantic level. It is independent of semantic concept type or subcomponents and treats all semantic components with equal importance.

TABLE II. DEFINITION OF BASIC METRIC ELEMENTS

Name	Definition
TP: True Positive	Total number of True Positives: The concept is present and correctly and fully (including all subcomponents) detected.
FP: False Positive	Total number of False Positives: The concept is incorrectly detected, i.e., either the concept is not present at all or one or more of its subcomponents are incorrectly detected.
TN: True Negative	Total number of True Negatives: The concept is correctly not detected, because the concept is not present.
FN: False Negative	Total number of False Negatives: A concept is not detected when it should have been.
TA: Total	Total number of annotated concepts, i.e., gold concepts.

We use the metric for both the performance for command extraction and also for callsign extraction. TABLE II. lists definitions that are the building blocks for the performance metrics. From the five building blocks we can derive *recognition rate* and *recognition error rate Eq (1,2)*. Additionally, we define in Eq.4 the *Fα-Scores* by defining *Recall* and *Precision* (Eq. 3).

$$RcR = Recognition\ Rate = \frac{TP + TN}{TA} \quad (1)$$

$$RER = Recognition\ Error\ Rate = \frac{FP}{TA} \quad (2)$$

$$Recall = \frac{TP}{TP + FN}; Precision = \frac{TP}{TP + FP} \quad (3)$$

$$F_{\alpha}Score = \frac{(1 + \alpha^2) * \text{Recall} * \text{Precision}}{(\alpha^2 * Precision) + Recall} \quad (4)$$

As observed from the F1-score in TABLE III. , the baseline ASR system fares quite poorly in detecting the rare words. Both performance values, WER as well as recall that are most important for correctly understanding the ATC message, are poor. Recall of ~0.05 roughly means that 95 (of 100) waypoints are substituted or deleted. Also, WER above 10% is significantly higher than observed on data such as those from HAAWAII project (3.1% from Table II in [5]). In the section IV, we discuss the proposed customization algorithms applied to the baseline ASR system to improve the performance of the rare words.

TABLE III. PERFORMANCE OF THE BASELINE ASR SYSTEM

Method	***WER [%]***	***Waypoint detection***		
		Precision	***Recall***	***F1-score***
Baseline	13.85	0.92	0.05	0.10

D. Baseline Performance on Semantic Level

Measuring an accuracy of ASR (typically done on word level) is not directly related to the problem of speech understanding (i.e., extraction of information on semantic level). As a complementary evaluation to ASR, we measure performance on semantic level using the metrics presented in the paper of Chen et al. [22] considering the commonalities of ASRU for ATC applications on both sides of the Atlantic [1]. The above data set in total consists of 1157 commands with 85 commands of type CONTACT as shown also for types CLIMB, DIRECT_TO and STATION in TABLE IV. The test set is same as the one described in section III-A. We did not show all command types, like HEADING, SPEED etc.

TABLE IV. SEMANTIC PERFORMANCE OF THE BASELINE SYSTEM

Baseline	Number	RecogRate	Precision	Recall	F1-Score
All Types	1157	58.3%	91.7%	61.2%	73.4%
CONTACT	85	36.5%	79.5%	39.7%	53.0%
CLIMB	90	90.0%	95.3%	94.2%	94.7%
DIRECT_TO	139	5.8%	47.1%	6.2%	10.9%
STATION	170	13.5%	85.2%	13.9%	23.8%

According to European annotation ontology [22] CONTACT result, e.g., from the utterance "*contact* ***boerde***". DIRECT_TO results from "*proceed to* **nienburg**" and STATION is the semantic interpretation of "*speed bird four one* ***maastricht*** *radar identified*". The airspace dependent words are marked in bold face. The mixture of command types is not representative for real life utterances. We added much more airspace dependent names to our test set. We see bad performance with recognition rates far below 50% for the CONTACT, DIRECT, and STATION command, whereas CLIMB is much better. It just consists of keywords, numbers, units, which are are not airspace dependent.

IV. CUSTOMIZATION OF ASR FOR NEW DOMAINS

This section describes the process of ASR adaptation or customization necessary to port the ASR engine to new domains or use cases. The main concept targeted by our work is to minimize the requirements for expert knowledge, allowing target users to customize the ASR technology on their premises. The process of customization can be done in several ways, as described below.

A. Adaptation using In-domain Data

The most obvious way of adapting the ASR is to use the concept of model adaptation by applying data-driven approaches leveraging set of speech and/or text transcripts from target domain. More specifically, the AM can be efficiently adapted to a new domain, e.g., to a target use-case airport/airspace, by exploiting some target data from the domain to fine-tuning the AM parameters and the LM statistics. In case of LM, the same data-driven concept can be easily applied, typically through the n-gram interpolation of an original LM with the one built from the target domain. However, both processes require certain expert knowledge, which is usually not part of an ATM personnel's skill-set. Thus, collaboration with the ASR developers is required. This also includes the collection of speech data sets from target domains, which could require certain amount of time. Additionally, the manual data transcription typically done by humans can take enormous amount of time. Furthermore, this process also requires certain unit-testing to make sure the new set of models perform well and the target models are not over-trained. Also, when porting the ASR system to a new domain, for instance DIAL project for the Celle sector, new waypoints are expected to be included.

B. Model Customization

The second method involves end-users customizing existing models to enhance the recognition of new words. In contrast to the aforementioned method, this paper presents an approach that enables the incorporation of a set of new words or word entities into the existing ASR framework without requiring additional manual transcriptions or retraining of existing models. Typically, the original ASR models are trained on huge data sets, but data at similar scales are unavailable during customization. Hence, this work tackles the problem in a more user-centric manner by providing the ATM personnel an easy-to-use utility. This utility lets ATM personnel to modify the ASR system by performing some simple pre-defined steps without requiring them to understand the background processes. Therefore, we have not delved further into the adaptation approach. Instead we have concentrated on strategies for customizing models to recognize new waypoints within the Celle Sector for "en-route" positions. The customization of the model in this study involves two distinct approaches: the first is referred to as G-boosting, and the second is termed Lattice-Rescoring. Both of these approaches will be explained in the following sections.

1) G-boosting

The G-boosting approach modifies the baseline LM trained for the ASR task. The trained LM is also stored as an FST (as mentioned previously). The LM is trained in such a manner that

it learns the most likely sequences or the context in which a word appears in the data. During decoding, the acoustic model information is used to predict many possible word sequences for a given test utterance. The LM provides the likelihood score for a predicted word sequence to be present in the current context. When there are many possible word sequences to select for a particular utterance, generally the one obtaining the highest LM score is selected. Although this is a very logical method to train and use the LM, it has its own disadvantages. The LM being a statistical model gets biased to the most frequent of the words seen during training data. In other words, a word seen many times during training will get higher LM score than a similar, but less frequent word. Thus, in cases where the acoustic model is not very confident about the predicted word in a particular context, the LM would tend to select the most frequent among the probable options. Our approach tries to solve this particular problem for the waypoints we are concerned about.

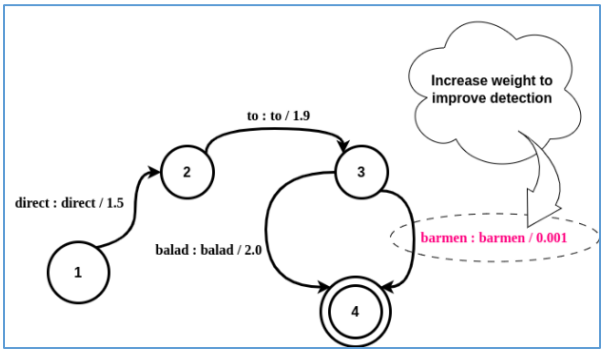

Figure 2. Word-boosting Operation in the Language Model.

Since the words are not seen many times during training, the final ASR output rarely predicts these words. To customize the ASR system to recognize these words, we update the weights in a pre-trained LM in a certain way, so that the likelihood of the new words being predicted increases. Fig. 2 illustrates a toy-example of the weight update step graphically. For instance, if the word "balad" is more frequently seen during LM training, it will be associated with a higher weight as compared to a less frequent "barmen". In such a scenario, to improve the detection of "barmen", we update the LM FST in such a manner that the word "barmen" also has a decent weight associated with it. The boosting factor is empirically decided based on the performance on a validation set. Basically, the LM FST discussed previously is a graph, where all the correct sequence of words is represented as arcs of the graph with their respective weights. Our approach searches the arcs in the FST corresponding to each of the new words and artificially boosts the weights of the words under consideration. The modified LM is then used to create a new decoding FST graph to replace the previous one. Subsequently, when the decoding is performed again using the new decoding FST graph, the new words are detected much better than earlier (see TABLE V.). The best part of this approach is that there is no need for expert intervention to perform this operation. An ATM personnel has to perform very basic steps (illustrated in Fig. 3). G-boosting was observed to perform quite well during experiments in improving the detection of rare words.

2) Lattice-Rescoring

As the name suggests, the second approach of Lattice-Rescoring is performed on the decoded word recognition lattices (mentioned previously). The lattices are a data-structure that store the top most likely set of decoded paths or word sequences for a given test utterance, along with their scores from the AM and the LM. In a normal decoding setup, the transcript for a given test utterance is the path that obtains the overall best score among all the possible sentences present in the corresponding lattice. As is the case with new and rare words discussed previously, even if they are present in one of the possible paths, they usually correspond to low LM scores and hence do not get selected as the best path. We observed that there is a scope to improve the recognition of rare words by modifying the decoded lattices and rescoring them. More specifically, we first create a small biased FST consisting of the set of rare words with boosted weights. Subsequently, a first pass decoding using the baseline LM is performed to obtain the initial lattices. These lattices are then composed with the biased FST in a manner that wherever the rare words are present in the decoded lattices, we update their corresponding LM scores with the boosted weight. After this operation, the modified lattice is used to compute the best path. The boost provided to the rare words' LM scores improves the chance of selecting the path that consists of the rare words. As with G-boosting, we avoid the need of expert intervention in performing Lattice-Rescoring as well.

The main distinction between G-boosting and Lattice-Rescoring is that Lattice-Rescoring works on-the-fly while generating the text transcripts of a speech utterance. Whereas G-boosting is performed in an offline manner before firing the ASR engine for generating the transcripts. Moreover, G-boosting is a permanent modification of the LM, whereas Lattice-Rescoring runs without permanently modifying the LM. In our experiments, we found Lattice-Rescoring improved the detection of rare words (see TABLE V.).

V. Results after Boosting the Waypoints

A. Word Level Performance

The performance of the word-boosting techniques is tabulated in TABLE V. As can be observed, the **G-boosting** technique improves the overall WER of the system from 13.85% to 9.55%. In terms of waypoint names detection, the baseline system detects 24 out of 443 occurrences whereas the detection improves to 256 out of 443 after performing G-boosting. Moreover, G-boosting improves the precision, recall and F1-scores of detecting the rare words from 0.92, 0.05, and 0.1 to 0.93, 0.55, and 0.69, respectively.

Lattice-Rescoring improves the precision, recall and F1-scores of detecting the rare words to 0.93, 0.09, and 0.17 respectively. Combining the G-boosting and Lattice-Rescoring methods improve the overall WER to 9.43%, while the rare word detection recall and F1-scores improve to 0.58 and 0.70 respectively. Interestingly, the overall WER increases to 14.04% with the Lattice-Rescoring approach, which is worse

than baseline. Also, the combination of G-boosting and Lattice-Rescoring reduces the precision of detecting the rare words. Such results indicate that the Lattice-Rescoring method introduces some false negatives in the modified output. In other words, the combination of both methods in rare cases cause over-boosting of the waypoints such that they may get detected in wrong places in the speech. A possible reason for such over boosting might be because this work applies the Lattice-Rescoring approach using a three-word context match around the rare words with a fixed boosting factor. A longer context would lead to a stricter match, thereby minimizing over boosting, but could also cause poorer recognition in genuine cases. In the future, we will try to optimize the context length and the boosting factor for Lattice-Rescoring. Nevertheless, the improvements obtained using these two methods is far more significant. Such results indicate the effectiveness of the proposed method in improving the detection performance of rare words in ASR.

TABLE V. PERFORMANCE OF THE CUSTOMIZATION TECHNIQUES

Method	***WER [%]***	***Waypoint detection***		
		Precision	***Recall***	***F1-score***
Baseline	13.85	0.92	0.05	0.10
G-boosting	9.55	**0.93**	0.55	0.69
Lattice-Rescoring	14.04	**0.93**	0.09	0.17
G-boosting + Lattice-Rescoring	**9.43**	0.88	***0.58***	**0.70**

B. Semantic Level Performance

TABLE VI. shows the performance for all command types (Column "All") and for the airspace dependent command types CONTACT, DIRECT_TO, and STATION for the different boosting techniques. The improvement of G-boosting and the combined technique is not only observed on word level, but also on the semantic level. A command is considered as recognized, if the callsign, the type, the value, the conditions etc. are correctly extracted from the recognized sequence of words. We see a dramatic improvement from baseline to the combined technique for CONTACT by almost 60% absolute and STATION with more than 75% absolute. Both command types have an F1-score around 95%. The location names that are uttered in connection with those two command types appear on a more regular basis and are of a low number so that it is easier to extract the semantics. The recognition rate of DIRECT_TO also improved by a factor of two, but on a very low level. The list of potential waypoints that could be correct seems to be too big in order to choose the correct one. A potential solution to look into in the future would be to take the decision about the target waypoint on semantic level knowing the aircraft trajectory and the likelihood of mentioning one of the waypoints from the N-best hypotheses.

TABLE VI. SEMANTIC LEVEL PERFORMANCE

	All		**CONTACT**		**DIRECT_TO**		**STATION**	
	RecRate	**F1**	**RecRate**	**F1**	**RecRate**	**F1**	**RecRate**	**F1**
Baseline	58.3%	73.4%	36.5%	53.0%	5.8%	10.9%	13.5%	23.8%
G-boosting	76.8%	**86.3%**	94.1%	96.4%	10.8%	19.5%	88.2%	93.5%
Lattice-Rescoring	59.1%	74.0%	35.3%	51.7%	9.4%	17.1%	21.8%	35.7%
G-boosting + Lattice-Rescoring	**76.8%**	**86.2%**	**95.3%**	**97.0%**	**11.5%**	**20.6%**	**89.4%**	**94.1%**

VI. STEPS TO BE TAKEN BY END-USERS

Thanks to our user-friendly solution proposed in this paper, the two previously mentioned word-boosting methods (G-boosting and Lattice-Rescoring) are designed for easy implementation, requiring minimal effort and expertise. More specifically, we developed scripts that empower anyone to make the necessary adjustments to the LM or the decoded word recognition lattices simply by providing a list of words in need of improvement. In cases, where these words are absent from the dictionary, it may be necessary to supplement the list with phoneme sequences representing those words. Beyond these considerations, end-users are not required to perform any additional complex steps when performing G-boosting. This process can be repeated as often as needed for any new set of words. Similarly, for the lattice-rescoring scripts, only the new words to be boosted are necessary as input. The scripts handle the remaining steps, including the creation of the biased FST, its combination with the decoded first-pass lattices, and the recalculation of the best path. The list of steps that an end-user must follow to utilize the methods mentioned above are represented pictorially in Fig. 3.

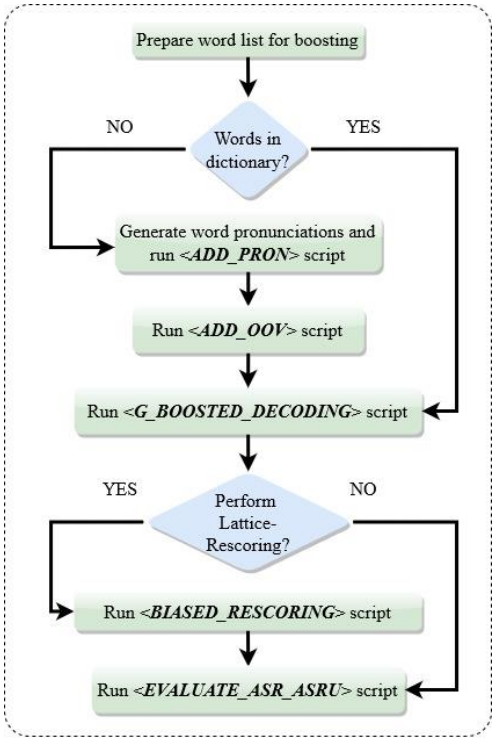

Figure 3. Steps to be followed by ATM personnel to perform ASR customization.

VII. CONCLUSIONS

This work describes two methods for customizing deployed ASR systems to new airports/airspaces to improve their performance in detecting seldom occurring terms such as waypoint or frequency position names that were rarely or never seen during training. The two methods are known as G-boosting and Lattice-Rescoring. In addition, we have developed scripts where these two methods are implemented in a user-friendly way that allows to perform the necessary adaptation without any expert knowledge of speech recognition. Such ease of use makes the methods an important utility if any user such as an ANSP needs to adapt their simulation environment or transfer operational environments including ASR functionality to new airports or airspaces or even for prototyping applications. The two methods are shown to provide significant improvement in detecting the rare words. G-boosting individually performs the best with more than 500% improvement in detecting rare words than the baseline system. Upon combining G-boosting and Lattice-Rescoring methods, we even obtain a relative improvement by a factor of 7 in detecting rare words, i.e., from an F1-score of 10% to 70%. Moreover, the command recognition rate of airport dependent names for different command types improve by a factor of two to three.

REFERENCES

[1] S. Chen, H. Helmke, R. Tarakan, O. Ohneiser, H. Kopald, and M. Kleinert, "Effects of Language Ontology on Transatlantic Automatic Speech Understanding Research Collaboration in the Air Traffic Management Domain," *Aerospace*, 10, 526, 2023.

[2] D. Schäfer, "Context-sensitive speech recognition in the air traffic control simulation," Doctoral dissertation, Universität der Bundeswehr München, Germany, 2000.

[3] I. Gerdes, M. Jameel, R. Hunger, L. Christoffels, and H. Gürlük, "The automation evolves: Concept for a highly automated controller working position," *33rd Congress of the International Council of the Aeronautical Sciences, ICAS 2022*, Stockholm, Sweden, 2022.

[4] M. Jameel, L. Tyburzy, I. Gerdes, A. Pick, R. Hunger, and L. Christoffels, "Enabling Digital Air Traffic Controller Assistant through Human-Autonomy Teaming Design," *42nd AIAA/IEEE Digital Avionics Systems Conference (DASC) 2023*, Barcelona, Spain, 2023.

[5] 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," *15th USA/Europe Air Traffic Management Research and Development Seminar (ATM2023)*, Savannah, GA, USA, 2023.

[6] H. Helmke, M. Kleinert, A. Linß, P. Motlicek, Hanno Wiese, L. Klamert et al., "The HAAWAII Framework for Automatic Speech Understanding of Air Traffic Communication," submitted to 13th SESAR Innovation Days, Seville, Spain, 2023.

[7] R. García, J. Albarrán, A. Fabio, F. Celorrio, C. Pinto de Oliveira, and C. Bárcena, "Automatic Flight Callsign Identification on a Controller Working Position: Real-Time Simulation and Analysis of Operational Recordings," *Aerospace*, 10, 433, 2023.

[8] H. Helmke, M. Kleinert, S. Shetty, O. Ohneiser, H. Ehr, H. Ariliusson, T. S. Simiganoschi, A. Prasad, P. Motlicek, K. Vesely, K. Ondrej, P. Smrz, J. Harfmann, and C. Windisch, "Readback error detection by automatic speech recognition to increase ATM safety," *14th USA/Europe Air Traffic Management Research and Development Seminar (ATM2021)*, Virtual Conference, 2021.

[9] J. Zuluaga-Gomez, A. Prasad, I. Nigmatulina, P. Motlicek, and M. Kleinert, "A Virtual Simulation-Pilot Agent for Training of Air Traffic Controllers," *Aerospace*, 10, 490, 2023.

[10] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Comput. Speech Lang*. 2002, 16, pp. 69–88.

[11] M. Mohri, F. Pereira, and M. Riley, "Speech recognition with weighted finite-state transducers," in *Springer Handbook of Speech Processing*; Springer: Berlin/Heidelberg, Germany, 2008, pp. 559–584.

[12] M. Riley, C. Allauzen, and M. Jansche, "OpenFst: An Open-Source, Weighted Finite-State Transducer Library and its Applications to Speech and Language," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Companion Volume: Tutorial Abstracts; Association for Computational Linguistics: Boulder, CO, USA, 2009, pp. 9–10.

[13] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," *Proc. Interspeech 2013*, 2013, pp. 2345–2349.

[14] H.A. Bourlard and N. Morgan, "Connectionist Speech Recognition: A Hybrid Approach," *Springer Science & Business Media*: Berlin/Heidelberg, Germany, 247, 1993.

[15] A. Srinivasamurthy, P. Motlícek, I. Himawan, G. Szaszák, Y. Oualil, and H. Helmke, "Semi-Supervised Learning with Semantic Knowledge Extraction for Improved Speech Recognition in Air Traffic Control," *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association*, Stockholm, Sweden, 2017, pp. 2406–2410.

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

[17] J. Zuluaga-Gomez, K. Veselỳ, I. Szöke, P. Motlicek, M. Kocour, M. Rigault, K. Choukri, A. Prasad, S.S. Sarfjoo, I. Nigmatulina et al, "ATCO2 corpus: A Large-Scale Dataset for Research on Automatic Speech Recognition and Natural Language Understanding of Air Traffic Control Communictions," arXiv 2022, arXiv:2211.04054.

[18] J. Zuluaga-Gomez, K. Veselý, A. Blatt, P. Motlicek, D. Klakow, A. Tart et al., "Automatic call sign detection: Matching air surveillance data with air traffic spoken communications," In *Proceedings*, 59, 14, MDPI, 2020.

[19] F. Jelinek, "Statistical methods for speech recognition," MIT press, 1998.

[20] 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 (ASRU)*. IEEE, 2011.

[21] H. Helmke, J. Rataj, T. Mühlhausen, Y. Oualil, M. Schulder, O. Ohneiser, H. Ehr, and M. Kleinert, "Assistant-Based Speech Recognition for ATM Applications," in 11th USA/Europe Air Traffic Management Research and Development Seminar (ATM2015), Lisbon, Portugal, 2015.

[22] S. Chen, H. Helmke, R. Tarakan, O. Ohneiser, H. Kopald, and M. Kleinert, "Effects of Language Ontology on Transatlantic Automatic Speech Understanding Research Collaboration in the Air Traffic Management Domain," *Aerospace 2023, 10, 526.*, 2023.