Thirteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2019)

Cost Reductions Enabled by Machine Learning in ATM

How can Automatic Speech Recognition enrich human operators' performance?

Hartmut Helmke,
Matthias Kleinert, Jürgen Rataj
German Aerospace Center (DLR),
Braunschweig, Germany
Hartmut.Helmke@dlr.de,
Matthias.Kleinert@dlr.de,
Juergen.Rataj@dlr.de

Petr Motlicek
Idiap Research Institute, Martigny,
Switzerland, Petr.Motlicek@idiap.ch

Dietrich Klakow
Saarland University (UdS),
Saarbrücken, Germany
Dietrich.klakow@lsv.uni-saarland.de

Christian Kern,
Austro Control, Vienna, Austria
Christian.Kern@austrocontrol.at

Petr Hlousek,
Air Navigation Services of the Czech Republic, Jenec, Czech Republic
Hlousek@ans.cz

***Abstract*—Various new solutions were recently implemented to replace paper flight strips through different means. Therefore, digital data comprising instructed air traffic controller (ATCO) commands can be used for various purposes. This paper summarizes recent works on developing speech recognition systems to automatically transcribe commands issued by air-traffic controllers to pilots allowing decrease of ATCOs' workload, which leads to significant increase of ATM efficiency and cost savings. First experiments in AcListant® project have validated that Assistant Based Speech Recognition (ABSR) integrating a conventional speech recognizer with an assistant system can provide an adequate solution. The following EC H2020 funded MALORCA project has proposed new Machine Learning algorithms significantly reducing development and maintenance costs while exploiting new automatically transcribed speech corpora. In this paper, besides recapitulating achieved recognition performance for Prague and Vienna approach, new statistics obtained from various error analysis processes are presented. Results are detailed for different types of ATC commands followed by rationales causing the performance drops.**

Keywords: Machine Learning, Assistant Based Speech Recognition, Unsupervised Learning, Command Prediction Model, Automatic Speech Recognition, MALORCA, Annotation, Transcription

I. INTRODUCTION

The steadily increasing air traffic creates more and more challenges concerning safety, capacity, efficiency, and environmental performance for air traffic management (ATM). Hence, these challenges as well as pressure on costs are the key drivers for future developments in ATM. Increasing digitization and automation is the widely accepted methodical answer to cope with them, as also addressed by SESAR and NextGen programs. The transfer of analogue data into digital formats is the central aspect of digitization. The digital formats itself are the starting point for each modern automation solution. Already today, a high degree of digitization is present in many ATM systems. The communication between air traffic controllers (ATCOs) and pilots, however, is excluded so far. It can be assumed that direct radio communication between controllers and pilots will continue in the next years despite the upcoming use of data links. Further, it can be assumed that in the transition phase both communication methods will exist for a considerable time. The content of this communication is of significant importance for the digital representation of the world in the automation systems (digital world). Therefore, the spoken commands must be digitized. This enables the digital world to include the contents of the communication between controllers and pilots into its situation monitoring, decision making processes, and post-operations analysis processes, which is also being addressed in the US, e.g. for using Automatic Speech Recognition (ASR) for better understanding of Performance-Based Navigation procedure utilization [1].

Nowadays, ATCOs are using mouse or keyboard explicitly entering the spoken commands into the digital systems to support the automation. This, however, leads to significant additional workload for them, which counteracts the goal of automation. By avoiding this unnecessary effort for the controller, the nascent cognitive resources could be used to guide the traffic more efficiently like initially intended with the introduction of automation.

Even if a perfect data link exists between controllers and pilots, it is questionable whether inputting commands by mouse and keyboard is the best way to fulfil this task. Since thousands of years, speech is the most natural way of human-to-human communication, it is reasonable to exploit the same type of communication to enhance human-machine interaction. Recent successes of Alexa, Siri and other voice assistants are strong hints that speech input may also be the best way for a human-to-machine communication, in which the machine world adapts to human capabilities and not vice versa.

Based on that insight, DLR and Saarland University (UdS) developed an Assistant Based Speech Recognition (ABSR) tool for ATCOs in the AcListant® project [2]. The project

MALORCA project and PJ 16-04 solution are partly funded by SESAR Joint Undertaking (Grant Numbers 698824 resp. 734141).

results showed that ABSR significantly reduces controllers' workload for system inputs [3] and increases ATM efficiency based on the released cognitive resources of the controller. This performance increase was demonstrated for Düsseldorf approach in a simulator environment. Significant fuel savings of 50 to 65 liters kerosene per flight were enabled by reducing the flight times, once the controllers were supported by ABSR and thus spent less time with manual inputs, but more time for reasoned guidance of aircraft was available [4].

To reach the full advantages of ABSR in reality, deployment in large scale is necessary. Controllers' cognitive resources are released with each deployment, which could be used to address the above-mentioned challenges. Determining factors for the number of deployed systems in ATM are the integration costs into existing ATM platforms and the maintenance costs. Deployment costs are still considerable high for ABSR, because ABSR must be adapted to each working environment, like an airport. Adjustments comprise specific waypoints, frequencies, deviations from standard phraseology or specific acoustic and semantic variability, like accented speech of ATCOs. So far the process of adaptation has requested significant involvement of experts. For the AcListant® project, which reached a competitive speech recognition performance, the required financial support was of about 1.3 Mio € to manually adapt and assess the technology for one approach, i.e. Düsseldorf. Hence, for the effectiveness of ABSR, it is of utmost importance that implementation as well as maintenance is simple and cheap, which includes further updates related to periodic changes in target domain.

The basic idea to overcome the need for manual adaptation was to develop an ABSR solution, which automatically adapts to specific environments by exploiting Machine Learning, i.e. data-driven, algorithms, because they are capable to offer much lower deployment and maintenance costs. SESAR Exploratory Research funded this idea in the Horizon 2020 project MALORCA (Machine Learning of Speech Recognition Models for Controller Assistance) [5].

The rest of the paper is written as follows: section II addresses related work in ASR domain for ATM applications. The active learning approach applied in MALORCA project is described in section III. Section IV gives a detailed overview of MALORCA's iterative training of three different models of ABSR. Section V analyzes the recognition results and discusses remaining problems with suggested solutions to overcome them. The last section concludes this paper

II. Related Work

A. Automatic Speech Recognition Application in ATM

Artificial Intelligence (AI) and in particular Machine Learning (ML) applications have made significant progress in the last few years, enabling computers to make a series of major breakthroughs that were previously impossible [6]. One of the successful "application" fields of ML is ASR, which has recently shown remarkable improvements in understanding human conversational speech. Speech recognition has developed quite independently for a very long time compared to the rest of Machine Learning community. Many interesting results were obtained in 90s, applying artificial neural networks into ASR [7]; especially the work by Mikolov et al. [8] in the area of language modeling and Seide et al. [9] on acoustic modeling have boosted the interest in neural networks in the speech community. However, neural networks require large training corpora and are thus difficult to apply to the ATM domain. Chen and Kopald used speech recognition to build a safety net for airport surface traffic to avoid aircraft using a closed runway [10]. They presented an approach to detect pilot read back errors in 2017 [11].

B. Machine Learning for Automatic Speech Recognition

Machine learning takes into account different types of learning: **supervised learning**, where labelled training data is available, **unsupervised learning**, where no labelled training data is available and **semi-supervised learning** combining both approaches. Either a speech recognizer can be treated as one big ML problem or it can be broken down into an ML problem for the so-called acoustic mode (AM)l and a separate ML problem for the so-called language model (LM). It is known for the acoustic modeling that techniques like MLLR (Maximum Likelihood Linear Regression) or MAP (Maximum a Posteriori) can be used in a semi-supervised setting with considerable success [12]. Recently, semi-supervised learning of NN based acoustic models for special areas such as YouTube videos was performed [13]. For language modeling, neither unsupervised nor semi-supervised learning has been very successful. Bellegarda describes adaptation of non-neural LMs [14]. Supervised adaptation of NN can be done either in a rescoring step [15] or directly by mapping the NN to a tree in the first pass of speech recognition decoding [16].

C. Assistant Based Speech Recognition (ABSR)

One promising approach to improve ASR performance is using context knowledge regarding expected utterances. These attempts go back to the 80s [17], [18]. This information may heavily reduce the search space and lead to fewer missed recognitions [19]. Oualil et al. [20] analyzed the benefits of using context information for pre-processing versus using context for post-recognition.

Helmke et al. extend the usage of context by generating the context from an assistance system, i.e. an AMAN, to support ABSR [2]. ABSR started with the study of Shore in 2011 [21]. In a pilot study with a limited set of callsigns and commands, Shore et al. [22] reported command (recognition) error rates below 5%. They used an acoustic model derived from the Wall Street Journal recognition corpus. In 2016, it was shown that ABSR significantly reduces controllers' workload, which translates into fuel burn reduction and an increased runway throughput. These results were quantified in [3] resp. [4]. MALORCA project aims at automatically adapting the speech recognition building blocks to different approach areas. Learning of command prediction, i.e. the relevant part of the assistant system, was described in [5]. Automatic adaptation results for Vienna and Prague approach area from controller-pilot speech recordings and the corresponding radar tracks were presented in [23]. Command recognition error rates of the baseline system were reduced from 7.9% to below 0.6% for Prague and from 18.9% to 3.2% for Vienna using each time 18

hours of untranscribed speech recordings without silence. The buildings blocks and their adaptation to different approach areas were presented in [24]. No safety issues were observed even if speech recognition failed. The controller detected all misrecognitions [25]. In October 2018, a speech recognition challenge was released by Airbus to develop ASR for an air traffic control scenario [26], allowing large variety of academy and industry to gain an access to real (i.e. manually transcribed) data and develop new ML algorithms in this domain. Although many applications achieved acceptable recognition performance with respect to word error rate, it became obvious that the lack of context information, which was not provided by the organizers, prevented from receiving comparable results possible with ABSR.

III. Building Blocks of Assistant Based Speech Recognition

Figure 1 shows a rough overview of the four main modules of an ABSR system, which are referred as DATA, TEXT, COMMAND and USER [24]: The DATA module generates and supplies the whole system with two types of data: dynamic and static. Dynamic data is represented by the voice input signal resp. an acoustic feature vector extracted from the signal and output of an assistant system (i.e. radar data, flight plan information, weather information, sequence data etc.). Static data for a given environment is represented by names of waypoints, runways, used frequency values etc.

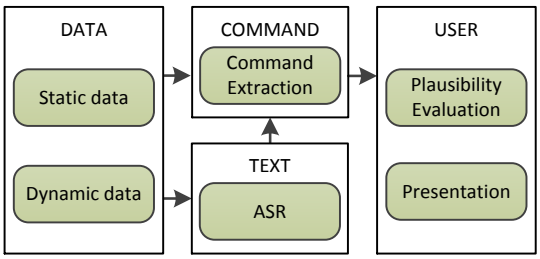

Figure 1. *Main modules of ABSR (taken from [24])*

The TEXT module uses some of the data provided by the DATA module and executes ASR related tasks on a given speech signal. This includes a Speech-to-Text conversion, i.e. the speech signal is transformed via feature extraction into a sequence of words. To do this, an **ASR decoder** transforms the acoustic feature vector X into a sequence of spoken words W = (w1, w2, w3 …), by applying the Bayes' theorem to find the word sequence, which maximize a posteriori probability P(W|X). The following three domain dependent models are used by ASR decoder:

- The **acoustic model** (AM) maps the input feature vector X of a phonetic unit (usually context-dependent phone) while taking into account the regional difference of speaking English (e.g. Czech English, or German English). Speaker independent or speaker dependent models can be applied. Deep Neural Networks (DNNs) are nowadays mostly used for acoustic modeling.
- The **lexicon** contains a list of all allowed words of the application domain together with their pronunciations. Phoneme sequences are mapped to a sequence of recognized words, while taking into account different pronunciations that may exist for the same word.
- The **Language Model** (LM) applies a probability distribution over a sequence of words to determine the most likely word sequence related to an audio input. Normally this task can be addressed by Context-Free Grammars (CFGs), since ATC commands are supposed to follow a relatively strict phraseology. However, ATCOs often deviate from standard phraseology and hence, ATC suggested CFGs are found too strict to learn all the deviations used by ATCOs. Instead, N-gram Statistical Language Models (SLMs) have shown significant improvements over CFG. The output generated by ASR decoder (i.e. output provided by TEXT module) can generate several word-strings (hypotheses). This process is referred to as **N-Best-Generator** that selects the N (e.g. N=5) most probable word sequences W (according to total likelihood given by ASR decoder), instead of extracting only the most probable sequences of words (N=1). More details are given in [24].

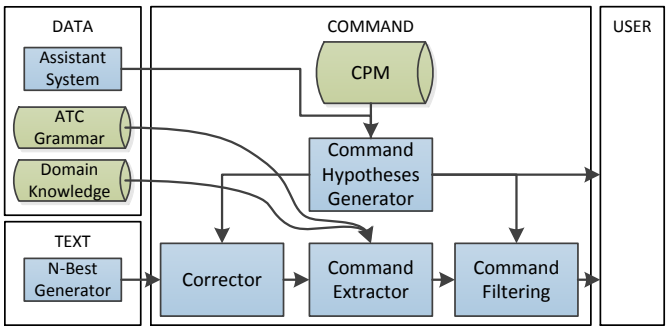

Figure 2. *Components and integration of COMMAND module (from [24])*

The main module COMMAND is used to convert the raw sequences of words obtained from the N-Best Generator of the TEXT main module to ATC commands; see Figure 2:

- The **Command Hypotheses Generator** generates a set of commands, which are plausible (with respect to information of assistant system of the DATA module) in the current air traffic situation.
- The **Corrector** modifies the recognized word sequences of the N-Best Generator by leveraging an output of the Command Hypotheses Generator. For instance a callsign: "*lufthansa alpha romeo*" might be replaced by "*lufthansa **one** alpha romeo*", if only a "DLH1AR" is in the air (i.e. found on radar).
- **Command Extractor** transforms the corrected sequence of words (e.g. "*good morning speedbird bravo one charly reduce two twenty or less*") to ATC commands (e.g. "*BWAB1C REDUCE 220 none OR_LESS*"; none specifies that the word "*knots*" for the unit was not spoken).
- The output of Command Extractor might still end up with multiple possible command sequences, if its input from N-Best-Generator contains more than one word sequence, because Command Extractor just transform word

sequences into command sequences. Different word sequences may result in the same command sequence.

- The **Command Filtering** block selects the most plausible command sequences generated from spoken command, while taking into account the set of possible commands generated from the radar situation by the Command Hypotheses Generator.

Besides these building blocks, the COMMAND module requires a **Command Prediction Model** (CPM). CPM contains rules to generate the set of possible commands for selected ATC approach. Details of CPM and its development using ML approaches can be found in [5] and [24].

The output may still not be unique, i.e. different command hypotheses could result from the same voice input. Finally, the USER main module selects a unique output, which is adequately presented to the controller. Plausibilities and the set of possible commands are used for this task. If the output after this process is not unique or none of the command hypotheses is plausible, no output is shown to the ATCO. Figure 3 summarizes the transformation of inputs from the speech signal and radar data into recognized commands finally presented on the ATCO's HMI. The bold blue arrows show the flow and transformation of the speech signal into ATCO commands and the thin black arrows show in which steps of the process radar data resp. predicted commands are used. Green blocks represent unique data elements, whereas pink blocks correspond to data elements, which can be ambiguous for the same input data.

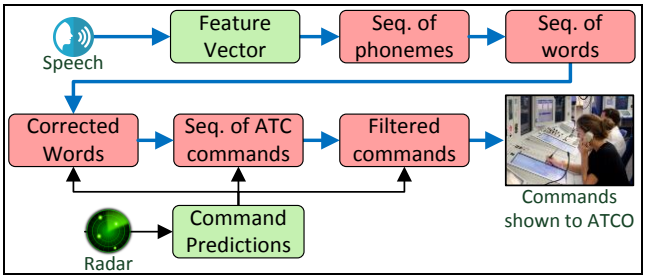

Figure 3. *Transformation of input data into output for ATCO*

IV. Iterative Training of the ABSR Models

This section describes the training of the three main models of ABSR. Training of the CPM is presented in the first subsection. The next two subsections describe the training of AM and LM. The last subsection details iterative improvement of all three models.

A. Command Prediction Model (CPM) training

A prediction area is modeled for each command type as shown by the dark hash symbols ('#') in Figure 4. Command types are e.g. *DESCEND*, *REDUCE*, *CLEARED ILS*, *HEADING LEFT*. A detailed analysis of the modeled command types is presented in section V. A set of predefined rules to each command type (e.g. IF flight type is arrival AND controller working position is Feeder AND speed > 220 knots) is defined. If the "Hypotheses Generator" detects that, a position (lat/long) of an aircraft is inside an area of a specific command type and the condition of the rule for this area is true, the command values related to that flight and command type are predicted for that aircraft. The values could be e.g. for descend 70, 80, 100, 3000, 3400 and 4000.

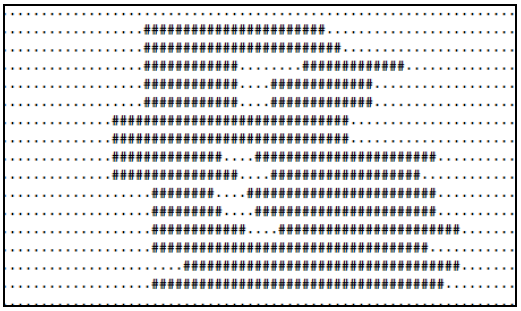

Figure 4. *Prediction area of a command type*

Each symbol in the prediction area (see [5]) represents a square of approx. 1 nm by 1 nm. These areas can be generated manually [27] or learned automatically from transcribed controller utterances and corresponding recorded radar data. These approaches require either expert knowledge for manual creation and/or expensive manual transcription work of recorded controller speech utterances. In order to remove the need of manual work, the approach of MALORCA project learns these areas from automatic transcriptions. For each controller utterance, the corresponding lat/long positions are known from the recorded radar data, but the correct controller commands are unknown. The only known things are the recognized ASR commands. Several heuristics are described in more detail in [5], [24] to filter out wrong recognitions: e.g. *CLIMB* commands for inbounds are seldom; *QNH* values do not change by ten hectopascals within five minutes; recognitions might be wrong, if the same command type is not observed in the vicinity.

B. Acoustic Model training

To develop an acoustic model, MALORCA project relies on open source out-of-domain English corpora used to initialize the training [33]. Most of available in-domain speech recordings are given in 8 kHz quality. Therefore, the same type of data is used over all acoustic modeling. Conventional technology combining deep learning, i.e. DNN, employed in Hidden Markov Modeling (HMM) framework is used. The technology, referred to as hybrid acoustic modeling, not only offers state-of-the-art performance, but also allows for rapid acoustic domain adaptation, which is essential for the ABSR system approach. It is used for (1) speaker-dependent modeling, (2) bootstrapping the model from rich resources (i.e. out-of-domain dataset) leveraging other ASR application domains and adapting the generic model to a target-domain and (3) iterative re-training: The ASR decoder in addition to word hypotheses provides confidence measures which can be used to assess quality of automatically generated transcripts related to new speech data. The fused confidence measure can be directly applied to select the relevant speech data from new speech corpora and iteratively re-train the hybrid acoustic model.

C. Training of the Language Model

Language modeling techniques like the grammar-based models provide a large set of rules to cover the phraseology

used by controllers, whereas the Statistical Language Models (SLMs) learn these rules automatically along with the deviations regularly made by controllers (assuming enough training data is available) and also adapt to these deviations in a more robust manner than a grammar-based model. The experimental work of MALORCA project continues exploring SLMs. Even though, SLMs have shown to perform better than Grammar-based models [28], the MALORCA project has raised a unique challenge combining both model types, as the initial amount of transcribed data was relatively small (< 4 hours). As this can lead to a poor coverage of ATM commands, MALORCA project alleviate this problem by leveraging the ICAO grammar [29] and constructing a hybrid SLM from this grammar and already trained SLM.

The grammar specifies the set of rules, defining the correspondence of command words to ATM concepts. These classes can then be used to build a class-based SLM [30], which has shown an improved ASR performance. Intuitively, this class-based LM allows overcoming the problem with lack of data by mapping everything to a class space. In this class space, correlations can be learned at a concept level; unlike the regular SLMs used earlier [31].

These class-based LMs and regular SLMs are linearly interpolated [32] to produce the final hybrid SLM. Eventually, this hybrid LM is converted to a first-pass decoding finite state transducer [33] and employed in the ABSR pipeline; see [34] for details of AM and LM training in MALORCA project.

D. Iterative Model Improvement

The last three sections assumed manually transcribed data used for both AM and LM training through deep learning architectures. Another work considered unlabeled data to be used for training. As matter of fact, CPM training can totally rely on unlabeled data, if an initial speech recognizer is available for automatic labeling of controller utterances to recognized commands type.

An enhanced speech recognizer will result in an improved CPM, which will also enrich the command hypotheses. Overall, this approach allows enriching training corpora and re-training both AM and LM by relying on the feedback from radar data as an additional sensor, i.e. to use the set of predicted commands, to decide whether an automatic transcription is a good or a bad training data set.

After developing an initial domain-independent ABSR system, MALORCA project has automatically re-trained these models for two target approaches: Prague and Vienna. More specifically, (1) a basic AM and LM were used to (2) automatically transcribe additional 25% of the speech recordings (approx. 4.5 hours). (3) Then the CPM is trained employing 10% of automatic transcriptions (approx. two hours). (4) The recordings of the 25% data set are subdivided into good and bad training data by exploiting information provided by CPM developed in step 3. (5) Both AM and LM are retrained, (6) automatically transcribing 50% of the speech recordings, i.e. approx. nine hours. (7) The CPM was trained with 25% of automatic transcriptions filtering out wrong recognitions in the 50% data set, and (8) repeated the previous steps, until all the training data for training all three models has been used. 18 hours were used for both Prague and Vienna.

E. Results of Iterative Training

Table 1, taken from [24], shows the results for Vienna approach. The second row (0%) in this table shows the results, when the three models (AM, LM, CPM) are trained without any untranscribed data for AM and LM training resp. using 10% for initialization of CPM model. The following rows show the results for the training with 25%, 50%, 75%, and 100% of the untranscribed data applied for all three models. The meaning of the columns is:

- Command recognition rate (**RR**): number of correctly recognized commands, which are not rejected by CPM, divided by the total number of given commands (#TgC). A command is correct, if callsign, command type, command value and qualifier, e.g. left/right, are correctly recognized.
- Command recognition error rate (**ER**): number of recognized commands, which were not spoken and not rejected, divided by #**TgC**,
- Pure command recognition rate (**PRR**): number of correctly recognized commands, without considering rejection by Command Filtering using CPM, divided by #TgC,
- Pure command recognition error rate (**PER**): number of recognized commands, which were not spoken, i.e. false recognitions, divided by #**TgC**,
- Command prediction error rate (**CpER**): number of commands included in gold (i.e. really given) commands, which were not predicted, divided by #**TgC**,
- Average number of predicted commands per aircraft and situation (**#NPC**).

TABLE 1: METRICS FOR ITERATIVE IMPROVEMENT OF VIENNA APPROACH

Amount of untranscribed data used	RR [%]	ER [%]	PRR [%]	PER [%]	CpER [%]	#NPC
0%	60.0	1.6	67.2	18.9	15.2	14
25%	80.2	3.5	84.0	7.4	6.7	29
50%	82.4	2.8	84.7	6.7	4.6	39
75%	84.2	3.0	85.6	7.0	3.5	47
100%	85.2	3.2	86.4	6.6	3.2	53

Results based on 4211 given commands from 3.84 hours of speech excluding silence, i.e. 21.1 hours of radar data time

The first columns (RR, ER) show the rates, when filtering by the CPM is applied, i.e. using ABSR. The column PRR and PER show the performance, if only ASR is used (P = pure), output of Command Filtering without second filtering in USER module is considered. The prize of second filtering becomes clear when RR is compared to PRR. RR is less than PRR, because filtering in USER module is not perfect, i.e. correct recognitions are also filtered out. On the other hand, the profit from ABSR is clearly illustrated by the comparison of ER and PER. ER is much less than PER, because most of the wrong outputs of ASR are filtered out. Table 2, taken from [24], shows the corresponding results for Prague approach area.

Command Recognition Error Rate ER increases in both tables due to the implementation of CPM training. The number of predicted commands increases with the size of available training data. Next section, however, will show that much more training data can overcome this limitation.

TABLE 2: METRICS FOR ITERATIVE IMPROVEMENT OF PRAGUE APPROACH

Amount of untranscribed data used	RR [%]	ER [%]	PRR [%]	PER [%]	CpER [%]	#NPC
0%	79.8	0.29	85.9	7.90	8.1	28
25%	90.2	0.32	93.7	2.2	4.4	45
50%	91.3	0.37	93.5	2.3	3.0	58
75%	91.7	0.45	93.6	2.4	2.5	67
100%	91.9	0.60	93.7	2.4	2.3	70

Results based on 5339 given commands from 4.69 hours of speech excluding silence, i.e. 25.7 hours of radar data time

The results also show that the learning curve of Vienna does not reach its saturation limits. Increasing the data size by a factor of two (from 25% to 50% and from 50% to 100%) still improves the values. RR increases by 2.2% (absolute) from 25% to 50% data size and again by 2.8% (absolute) from 50% to 100%. Extrapolating the currently available 100% data by a factor of eight, an RR of 90.2 % for Vienna seems to be possible. Performing the same data extrapolation also for Prague with eight times more data, the recognition rate for Prague could reach 92.6%. It seems the trained Prague models are already close to saturation, i.e. the currently available 100% of learning data would be already sufficient. The next section, however, shows that also for Prague more training data will improve recognition performance for some command types.

V. EVALUATION OF MODEL TRAINING

As a first step, the performance of ABSR, concerning only using ASR part, is analyzed for different horizontal command types in Table 3.

A. Analyzis of Horizontal Command Type Recognition

Average command recognition rates for *HEADING* and *MAINTAIN HEADING* outperform average rates calculated over all command types. The average rates for *DIRECT_TO* commands are below the global average. More interesting, however, are poor recognition rates for *NAVIGATION_OWN* and *TURN* commands. 4,211 commands result from 3.8 hours of data without silence, 5339 result from 4.7 hours (Prague).

TABLE 3: PURE ASR RATES FOR HORIZONTAL COMMANDS

Command Type	Vienna			Prague		
	TgC	PRR [%]	PER [%]	TgC	PRR [%]	PER [%]
All commands	4211	86.4	6.6	5339	93.7	2.4
DIRECT_TO	400	79.8	12.8	346	87.3	4.6
HEADING	250	97.2	3.6	439	95.4	3.6
MAINTAIN HEADING	33	97.0	3.0	173	95.4	4.0
NAVIGATION OWN	7	71.4	57.1	4	75.0	625
TRANSITION	60	76.7	8.3	0		
TURN	21	38.1	0	4	0.0	0.0
TURN_BY	6	83.3	0	3	0.0	0.0

Green marks results, which are much better than average and red, which are much worse

One explanation for the poor rates is that only few samples of 11 respectively 25 are available for training. That, however, does not explain PER of 625% for *NAVIGATION_OWN*. The rates are calculated as following:

- If a command type is given and correctly recognized, which includes that callsign, type value etc. are correct, and it is not rejected, it is counted as correct recognition.
- If a spoken command is not recognized, it is counted either as an error or as a rejection for that recognized type.

For example, if the commands "*REDUCE 210 kt*" and "*DIRECT_TO PR530*" are given by the ATCO, but the recognition is "*REDUCE 120 kt*" and "*NAVIGATION_OWN*", assuming that "*REDUCE 120 kt*" is not predicted by CMP, command type *REDUCE* is rejected and command type *NAVIGATION_OWN* is an error. No error resp. rejection is counted for *DIRECT_TO*. This explains the very high PER for type *NAVIGATION_OWN*. It is often recognized, but seldom given by ATCO.

When analyzing the recognition results, it was observed that most of the recognition results of type *NAVIGATION_OWN*, were correctly recognized by the ABSR, but the commands were differently interpreted by different annotators. The utterance "*speedbird eight six one resume own navigation proceed direct rapet*" was annotated as shown in Table 4.

TABLE 4: DIFFERENT ANNOTATIONS OF DIFFERENT ANNOTATORS

Annotator	Command	Command
1	BAW861 DIRECT_TO RAPET	
2	BAW861 OWN_NAVIGATION	BAW861 DIRECT_TO RAPET
3	BAW861 OWN_NAVIGATION	
4	BAW861 OWN_NAVIGATION	DIRECT_TO RAPET

These annotations are neither correct nor wrong. Unique rules for annotation are needed to enable exchange of data of different annotators or even between different approach areas. Such a solution cannot be provided by researchers alone, because detailed and specific domain knowledge as well as the formation of a joint view on that problem is necessary. Only a broad industrial consortium is in the position to create an appropriate solution. In this case, the structure of SESAR with its exploratory and industrial research part plays an important role. As several partners of the MALORCA project are part of the exploratory as well as of the industrial research part of SESAR, they are in the position to bridge the gap between research and industry by aligning the work between parts in SESAR and speed up in this way the deployment of new technologies in ATM.

B. Unique Rules for Command Annotation

The SESAR 2020 funded solution 16-04 agreed on an ontology, i.e. unique rules, for command transcription and annotation [35]. The main elements of the ontology, agreed by 16-04 partners, are callsign and instruction. 16-04 partners include Air Navigation Service Providers (ANS CR, Avinor, Austro Control, DFS, LFV, NATS, Romatsa), Research Institutes (CRIDA, DLR), ATM supplier industry (Frequentis, Indra, and Thales) and Integra as ATM consultancy.

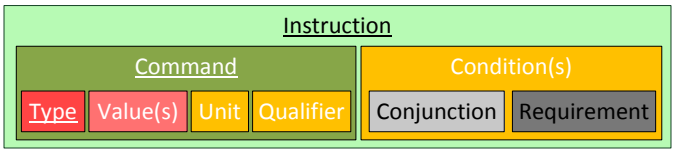

Figure 5. *Elements of an instruction of a clearance (taken from [35])*

Figure 5 shows that an instruction always consists of a command (darker green part is mandatory) and one or more optional (orange) conditions. A command is composed of a type, one or more values and an optional unit, like *FL ft* or *none*. Then an optional qualifier follows, like *LEFT, RIGHT, OR_LESS, BELOW.*

TABLE 5: ALL HORIZONTAL COMMAND TYPES OF ONTOLOGY

Command Type	Value(s)	Qualifier
TRANSITION	TransitionName	
DIRECT_TO	Waypoint(s)	
FOLLOW_ROUTE	RouteName	
HEADING	HeadingValue3 / HeadingString	LEFT/RIGHT/none
MAINTAIN_HEADING	HeadingValue3	
TURN_BY	HeadingValue1-2	LEFT/RIGHT/none
TURN		LEFT/RIGHT/none
CONTINUE_PRESENT_HEADING		
NAVIGATION_OWN		
STRAIGHT_IN_TURN		
CROSS_WP		Alt-Qualifier (optional)

An utterance may consist of multiple instructions for the same callsign or even for different callsigns. The callsign is always added to the instruction (or *NO_CALLSIGN*) independent of being repeated by the controller. Table 5 shows the different command types for horizontal commands. *TransitionName*, *WaypointName* etc. are airport depending knowledge. *HeadingValue3* is a value between one and 360 consisting of exactly 3 digits, e.g.. 005, 065, 210. *HeadingString* is one of the values "*NORTH*", "*EAST*", "*SOUTH*", "*WEST*" and "*RUNWAY_ DIR*". More details can be found in [35].

TABLE 6: APPROACH TYPES SELDOM USED FOR PRAGUE AND VIENNA

Command Type	How often observed	
	Vienna	Prague
CLEARED NDB / RNAV	0	2
INTERCEPT_LOCALIZER	10	7
CANCEL IFR	8	2
SQUAWK	9	6
TURN LEFT/RIGHT/none	21	4
TURN_BY LEFT/RIGHT/none	6	3
EXPECT RUNWAY	1	2
REPORT_NOW	1	2
INCREASE OR_LESS/OR_GREATER etc.	0	1
REDUCE_MIN_CLEAN_SPEED	2	10
STOP_CLIMB / STOP_DESCEND	5	1
RATE_OF_CLIMB OR_LESS/OR_GREATER etc.	12	0
RATE_OF_DESCENT OR/LESS/OR_GREATER etc.	12	0

Numbers based on manual annotation of test data presented in Table 1 and Table 2

The annotations of the MALORCA project were all automatically transformed into the new defined format. 24 command types of the ontology, which are used by approach controllers, were observed neither in Vienna nor in Prague airspace. This includes types like *CANCEL GNSS, CANCEL CLEARANCE, HOLDING, FOLLOW_ROUTE, REDUCE_BY, HIGH_SPEED_APPROVED*, and *EXPEDITE PASSING*.

Command types, shown in Table 6, were used, but occur very seldom in the transcribed 3.8 hours from Vienna (4,211 commands) resp. 4.7 hours from Prague (5339 commands). No reliable model training was possible for these types. Therefore, they are excluded from evaluations in the following sections.

C. *Recognition Performance for Different Command Types*

The 22 commands, shown in Table 7, were considered. Rows marked in yellow were excluded from further evaluations, because these command types were seldom used for that approach area and, therefore, learning data is not sufficient. Commands marked in blue were excluded, because their annotations are not reliable due to missing annotation rules when command annotation was done in late 2016. Cells marked in red, contain worse results compared to average results.

TABLE 7: METRICS FOR MOST RELEVANT COMMAND TYPES

Command Type	Vienna					Prague				
	Ttl	User Rates in %		ASR Rates in %		Ttl	User Rates in %		ASR Rates in %	
		Rec	Err	Rec	Err		Rec	Err	Rec	Err
CLEARED ILS	130	95	7	95	8	193	91	2	96	4
CONTACT	563	86	2	87	4	529	94	0	95	1
CONTACT_FREQUENCY	561	92	3	92	5	525	92	4	93	5
HEADING (LEFT, RIGHT, none)	250	91	1	97	4	439	90	0	95	4
MAINTAIN HEADING, CONTINUE PRESENT_HEADING	33	97	3	97	3	173	95	0	95	4
DIRECT_TO	400	76	8	80	13	346	80	1	87	5
NAVIGATION_OWN	7	0	14	71	57	4	75	200	75	625
TRANSITION	60	75	3	77	8	0				
EXPECT ILS	132	85	10	87	17	3	0	0	100	4667
INFORMATION QNH	204	89	5	89	5	220	98	0	98	0
INFORMATION ATIS	38	0	0	76	8	177	88	0	89	3
INIT_RESPONSE	270	0	0	85	21	523	97	0	97	1
NO_CONCEPT	254	72	0	72	0	349	84	0	84	0
REPORT ESTABLISHED	8	100	0	100	0	86	99	0	99	0
REDUCE (opt. OR_LESS, OR_GREATER)	126	96	0	97	2	205	91	2	92	5
SPEED (opt. OR_LESS, OR_GREATER)	207	86	1	87	5	121	79	2	82	8
MAINTAIN SPEED, (opt. OR_LESS, OR_GREATER) CONTINUE PRESENT_SPEED	16	31	6	44	6	20	85	0	85	0
NO_SPEED_RESTRICTIONS	45	80	4	80	4	172	95	0	95	0
DESCEND (opt. OR_ABOVE, OR_BELOW)	717	85	1	86	6	932	97.5	0.2	98	1
CLIMB (opt. OR_ABOVE, OR_BELOW)	335	89	3	89	7	275	96	0	98	1
ALTITUDE (opt. OR_ABOVE, OR_BELOW)	26	54	12	58	42	5	0	40	20	160
MAINTAIN ALTITUDE, PRESENT_ALTITUDE	15	60	27	60	47	15	80	0	80	13
Sums / Average	4397	79	3	87	7	5312	92	1	94	6

Command types marked in green in first column in Table 7 show the important commands, like *CLEARED ILS* or *DESCEND*. A recognition failure of these types might result in a safety issue. , Therefore, they are manually maintained in the radar label by the ATCO, if no ASR is available. The recognition rates of *MAINTAIN SPEED* (31% / 85%), *ALTITUDE* (54% / 0%) and *MAINTAIN_ALTITUDE* (60% /

80%) are below the average recognition rates. This should be solved, when more training data for these command types are available. More interesting are the problems of *DIRECT_TO* for both areas (76% resp. 80%) and the of *SPEED* command for Prague (79%).

These rejections and errors related to speed command types were analyzed in more detail. Results are:

1. Prague controllers use the phraseology "*start reducing*" or "*start reducing speed*" followed by a speed value. The sequence of words was mostly recognized, but this was never recognized as a *REDUCE* command, because these commands were never observed in the transcribed training data. Prague controllers used this command type six times in manually transcribed test data. It occurred 24 times in automatically transcribed data, which was not used for training Command Extractor. Vienna controller did not use this phraseology at all;
2. "*reduce speed **two zero zero** knots*" was often recognized with an additional word "***two***", i.e. " *reduce speed **two two zero zero** knots*". The controllers did not say, "*reduce to*". They did not hesitate after the "*reduce*". It was only observed for Prague data. The heuristic "if recognized speed value is greater than 1999 and first digit is two, then delete first digit" eliminates 12 speed recognitions errors without increasing recognition errors of speed commands;
3. Three times a speed command value was not predicted. Value 140 was not observed in training data, two times an overflight type was extracted from flight plan data, but it was an arrival;
4. One time "*reduce to two zero*" instead of "*reduce two two zero*" was recognized;
5. Four times a *SPEED* value was not predicted. The *SPEED* command was an *INCREASE* for an inbound, which was not expected;
6. Five times seldom used phraseology results in recognition of *NO_CONCEPT* (e.g. "*speed if you wish two three zero*", "*standard arrival routing speed two five zero*");
7. Four times a *SPEED* command was recognized as *REDUCE*, which was intended by the controller, but not said;
8. Three times no callsign was said at all (e.g. "*two eighty or more is fine*");
9. Four times the annotation was wrong: *MAINTAIN_SPEED* was recognized and said, but annotated was just *SPEED*.

The problems described in 1, 3, 5, 6, 7, 8 can be solved with more training data. Problem 9 requires better training of annotators with the annotation rules. Problem 2 is a software bug. Analysis of wrong recognitions of *DIRECT_TO* command results in the following observations:

1. The callsign was not said and, therefore, the whole *DIRECT_TO* command was not recognized in one case;
2. Command extraction from seldom used phraseology failed: "confirm proceeding to waypoint …";
3. "*is waypoint*", was once in test data, never in untranscribed trainings data;
4. "*waypoint is approved*" or "*it's approved waypoint*": observed once each in test data;
5. "*via waypoint*", 12 times in test data, 43 times in untranscribed trainings data. This was modeled neither in grammar nor in SLM.

If this phraseology is frequently used, more transcribed trainings data will solve the problem. Otherwise, it is a minor problem, which might be true for the first four ones.

6. The phraseology "*own navigation to waypoint*" was observed 12 times in test data and 21 times in untranscribed trainings data;
7. "*turn left/right direct to waypoint*" was observed 7 times in test data and 32 times in untranscribed trainings data.

The solution to both problems was already touched in Table 4, i.e. inconsistent manual command annotations fails to learn the phraseology from transcribed data. Using the ontology will solve the problem in the future.

Another problem with *DIRECT_TO* commands are seldom used waypoints. Either the waypoints *PR511, POLOM, ODNEM, PISAM, AKEVA, MAPIK, ULGIL, LOBMA, PR522, TOMTI,* and *BERVA* were not observed in the trainings data. The waypoint VLM, spoken as "*vlasim*", occurs 20 times in untranscribed word sequences, but the mapping from "*vlasim"* to VLM failed, i.e. was not modeled. The same applies for *FAF24*. The controller used the word sequences "*final approach fix runway two four*" or "*final approach fix two four*". The letter ICAO codes *LKMH, LKBU, LKBE, LHHK*, and *LOWS* do not occur in untranscribed training data. They seldom occur and their mapping to the airports *Mnichovo Hradiste, Bubovice, Benesov, Hradec Kralove*, and *Salzburg* is not modeled.

Currently CPM can predict only waypoints, which were also used in the past, i.e. which occur in untranscribed trainings data. This requires that waypoint must be said in trainings data and even more important the mapping from word sequence (e.g. Salzburg) to ICAO code used in *DIRECT_TO* as value (e.g. *LOWS*) must be modeled in static data (see Figure 1).

VI. CONCLUSIONS

This paper integrates the results from the two Horizon 2020 EC funded projects, which deal with Automatic Speech Recognition (ASR) applied in ATM world. MALORCA project has demonstrated successful applicability of Machine Learning (ML) of ASR to automatically transcribe ATCO utterances. Even with small amounts of ATC speech recordings MALORCA results reveal that command recognition rates above 92% are feasible. MALORCA project has exploited about four to five hours of transcribed and 18 hours of untranscribed speech data for both test airports (Prague and Vienna). In terms of overall human effort, the developed ML

algorithms have significantly reduced the manual data transcription effort. High speech recognition accuracies achieved in MALORCA indicate capabilities of ML algorithms to adapt generic ABSR systems to different deployment areas.

Although both ASR recognition accuracies, i.e. 92% of command recognition rate for Prague and 85% for Vienna, are high, further improvements are expected to increase the technology readiness level of ASR technology applied in ATC. The difference between Vienna and Prague datasets can partially be explained by noisy Vienna input data. Detailed analysis has shown that the "teacher" (human annotation) is still problematic, i.e. the manual annotations of ATCOs' utterances were sometimes ambiguous. Different experts annotated the same sequence of words differently. This problem was first discovered in the second half of the MALORCA project, when most of the annotation work was performed. In spite of MALORCA SESAR 2020 funded solution 16-04 agreed on an ontology, i.e. unique rules, for command transcription and annotation. 16-04 project includes partners from European Air Navigation Service Providers, research institutes, ATM supplier industry and consultancy. In many cases, when manual human command annotation was not correct, the output of ASR system, developed by MALORCA was nevertheless correct, i.e. the "real" command recognition rate of MALOCA modules is expected to be higher (approx. one percent) than reported earlier. A detailed analysis shows that the ASR performance is in fact higher than measured due to many errors introduced by humans into annotations. A ~1% improvement is estimated (e.g. from 92% to 93% in command recognition rate for Prague) if correct ground truth will be used for training and evaluation of the output of ASR system.

Analysis results also show that speech recognition rates of vertical commands (e.g. descend), being the most important ones with respect to safety, are very good, for instance a command recognition rate of 97.5% (with an error rate of 0.2%) was achieved on Prague data. These rates on the other hand require more than 930 training examples for the descend command type. Number of training data also explains the poor recognition rate of "*maintain heading*" type (only 16 training examples result in a recognition rate of only 31% for Vienna approach). This paper has brought a conclusion (known for ASR community for a long time) that increasing the amount of data improves recognition performance. Harmonization of command annotation as proposed by 16-04 ontology will ease the exchange of training data and, therefore, increase the amount of available learning data.

Note: ATCOs certainly remain responsible for the correctness of all entered values. They, therefore, require an applied routine to be as simple as possible allowing the check and in case of false recognitions the correction of an input, **before** it is processed by the system. Currently applied forms to enter command values in many cases trigger substantial status-changes without any simple "undo"-option. Considering this indispensable requirement to permanently monitor all proposed inputs, ATCOs would even significantly benefit from 92% resp. 85% recognition accuracies, even if a one click for confirmation is required. Therefore, in the context of SESAR 2020 Wave 2, it is planned to perform real-time OPS-room validations at the Vienna Approach Control Unit. Validations for a future usage at Tower Units are initially planned to be conducted on a virtual/remote-Tower platform at DLR in Germany. These validations will be supported by COOPANS and all MALORCA partners.

ACKNOWLEDGMENT

We thank all the controllers who anonymously provided us with real speech command examples. We also thank Youssef Oualil and Ajay Srinivasamurthy who contributed to MALORCA project. Youssef is now with Apple, Germany, and Ajay with Amazon.com, Bangalore, India.

REFERENCES

[1] S. Chen, H. Kopald, R. Tarakan, G. Anand, and K. Meyer, "Characterizing National Airspace System Operations Using Automated Voice Data Processing: A Case Study Exploring Approach Procedure Utilization," in 13th USA/ Europe Air Traffic Management Research and Development Seminar (ATM2019), Vienna, Austria, 2019.

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

[3] H. Helmke, O. Ohneiser, Th. Mühlhausen, and M. Wies,."Reducing controller workload with automatic speech recognition", in IEEE/AIAA 35th Digital Avionics Systems Conference (DASC). Sacramento, California, 2016.

[4] H. Helmke, O. Ohneiser, J. Buxbaum, and Chr. Kern, "Increasing ATM efficiency with assistant-based speech recognition", in 12th USA/Europe Air Traffic Management Research and Development Seminar (ATM2017). Seattle, Washington, 2017.

[5] M. Kleinert, H. Helmke, G. Siol, H. Ehr, M. Finke, A. Srinivasamurthy, and Y. Oualil, "Machine learning of controller command prediction models from recorded radar data and controller speech utterances," 7th SESAR Innovation Days, Belgrade, 2017.

[6] AlphaGo https://www.blog.google/technology/ai/alphago-machine-learning-game-go/, n.d.

[7] H. Bourlard, N Morgan, "Connectionist speech recognition: a hybrid approach", Kluwer academic publisher, 1994.

[8] T. Mikolov, M. Karafat, L. Burget, H. Cernocky, and S: Khudanpur, Sanjeev, "Recurrent neural network based language model ", Eleventh Annual Conference of the International Speech Communication Association, 2010.

[9] F Seide, G Li, and D Yu, "Conversational speech transcription using context-dependent deep neural networks", in 29th International Coference on Machine Learning (ICML'12), 2012.

[10] S. Chen and H. Kopald, "The Closed Runway Operation Prevention Device: Applying Automatic Speech Recognition Technology for Aviation Safety," in 11th USA/ Europe Air Traffic Management Research and Development Seminar (ATM2015), Lisbon, Portugal, 2015.

[11] S. Chen, H.D. Kopald, R. Chong, Y. Wei, and Z. Levonian, "Read Back Error Detection using Automatic Speech Recognition," in 12th USA/ Europe Air Traffic Management Research and Development Seminar (ATM2017), Seattle, WA, USA, 2017.

[12] H. Botterweck, "MAP defined by eigenvoices for large vocabulary continuous speech recognition", in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'01), 2001, pp. 353-356.

[13] H. Liao, E. McDermott and A. Senior, "Large scale deep neural network acoustic modeling with semi-supervised training data for YouTube video transcription", IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, 2013, pp. 368-373.

[14] J. Bellegarda, "Statistical language model adaptation: Review and perspectives", Speech Communication, 2004.

[15] X. Chen, T. Tan, X. Liu, P. Lanchantin, M. Wan, M.J.F. Gales and P. Woodland, "Recurrent Neural Network Language Model Adaptation For Multi-Genre Broadcast Speech Recognition", ISCA Interspeech, Dresden, 2015.

[16] M .Singh, Y. Oualil, D. Klakow, "Approximated and domain-adapted LSTM language models for first-pass decoding in speech recognition", in Proceedings of the 18th Annual Conference of the International Speech Communication Association (INTERSPEECH), Stockholm, Sweden, September 2017, pp. 2720-2724.

[17] S.R. Young, W.H. Ward, and A.G. Hauptmann, "Layering predictions: Flexible use of dialog expectation in speech recognition," in Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI89), Morgan Kaufmann, 1989, pp. 1543-1549.

[18] S.R. Young, A.G. Hauptmann, W.H. Ward, E.T. Smith, and P. Werner, "High level knowledge sources in usable speech recognition systems," in Commun. ACM, vol. 32, no. 2, Feb. 1989, pp. 183-194.

[19] D. Schäfer, "Context-sensitive speech recognition in the air traffic control simulation," Eurocontrol EEC Note No. 02/2001 and PhD Thesis of the University of Armed Forces, Munich, 2001.

[20] Y. Oualil, M. Schulder, H. Helmke, A. Schmidt, and D. Klakow, "Real-Time Integration of Dynamic Context Information for Improving Automatic Speech Recognition," Interspeech, Dresden, Germany, 2015.

[21] T. Shore, "Knowledge-based word lattice re-scoring in a dynamic context," Master Thesis, Saarland University (UdS), 2011.

[22] T. Shore, F. Faubel, H. Helmke, and D. Klakow, "Knowledge-Based Word Lattice Rescoring in a Dynamic Context," Interspeech 2012, Sep. 2012, Portland, Oregon.

[23] M. Kleinert, H. Helmke, G. Siol, H. Ehr, A. Cerna, C. Kern, D. Klakow, P. Motlicek et al., "Semi-supervised Adaptation of Assistant Based Speech Recognition Models for different Approach Areas", in IEEE/AIAA 37th Digital Avionics Systems Conference (DASC). London, England, 2018.

[24] M. Kleinert, H. Helmke, H. Ehr, Chr. Kern, D. Klakow, P. Motlicek, M. Singh, and G. Siol, "Building Blocks of Assistant Based Speech Recognition for Air Traffic Management Applications". 8th SESAR Innovation Days, Salzburg, 2018.

[25] M. Kleinert, H. Helmke1, G. Siol, H. Ehr, D. Klakow, M. Singh, P. Motlicek, Chr. Kern, A. Cerna, and P. Hlousek, "Adaptation of Assistant Based Speech Recognition to New Domains and its Acceptance by Air Traffic Controllers" in Proc. of the 2nd International Conference on Intelligent Human Systems Integration (IHSI 2019): Integrating People and Intelligent Systems, Feb. 2019,San Diego, California, USA.

[26] 2018 AIRBUS Air Traffic Control Challenge Workshop: https://www.irit.fr/recherches/ SAMOVA/pagechallenge-airbus-atc-workshop.html

[27] M. Hössl, H. Helmke, and J. Gottstein, "Why controllers seldom stick to the book and how their commands are predictable nevertheless," in ICRAT conference, Istanbul, May. 2014.

[28] Y. Oualil, D. Klakow, G. Szaszák, A. Srinivasamurthy, H. Helmke, P. Motlicek, "Context-aware speech recognition and understanding system for air traffic control domain", in IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2017), Okinawa, Japan, Dec. 2017, pp. 404-408.

[29] Eurocontrol, "All clear phraseology manual", Brussels Belgium, April 2011.

[30] F. Jelinek, and R. L. Mercer, "Interpolated estimation of Markov source parameters from sparse data",.in Proceedings of Workshop on Pattern Recognition in Practice, 1980.

[31] Y. Oualil, M. Schulder, H. Helmke, A. Schmidt, and D. Klakow, "Real-time integration of dynamic context information for improving automatic speech recognition," Interspeech, Dresden, Germany, 2015.

[32] M .Singh, Y. Oualil, D. Klakow, "Approximated and domain-adapted LSTM language models for first-pass decoding in speech recognition", in Proceedings of the 18th Annual Conference of the International Speech Communication Association (INTERSPEECH), Stockholm, Sweden, September 2017, pp. 2720-2724.

[33] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning: data mining, inference, and prediction," New York: Springer, 2009, pp. 485–586.

[34] A. Srinivasamurthy, P. Motlicek, I. Himawan, G. Szaszák, Y. Oualil, and H. Helmke, "Semi-supervised learning with semantic knowledge extraction for improved speech recognition in air traffic control," in INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm Sweden, Aug. 2017.

[35] H. Helmke, M. Slotty, M. Poiger, D. F. Herrer, O. Ohneiser et al., "Ontology for transcription of ATC speech commands of SESAR 2020 solution PJ.16-04," in IEEE/AIAA 37th Digital Avionics Systems Conference (DASC). London, United Kingdom, 2018.

AUTHOR BIOGRAPHY

Hartmut Helmke received his Diploma degree in Computer Science from the University of Karlsruhe (Germany) in 1989 and his doctor degree (PhD) from the chemical engineering faculty of the Technical University of Stuttgart in 1999. In 1989, he joined DLR's Institute of Flight Guidance in Braunschweig. He led the AcListant®, AcListant®-Strips and MALORCA project. Since 2017, he is the lead of the Automatic Speech Recognition activity within SESAR 2020 solution PJ16-04. Prof. Helmke is an assistant professor for Computer Science since 2001.

Petr Hlousek received his Diploma degree in Civil Aviation from the University of Transportation Zilina (Slovakia) in 1992. In 1993, he joined the Czech Airports Administration and worked at Prague Airport management in different positions up to 2014. From 2016, he works for the Air Navigation Services of the Czech Republic as Project manager where he is responsible for the ANS CR participation in PJ.16-04 ASR project.

Christian Kern joined Austro Control in 1997. He is the Air Traffic Management Director of Operations. Between 2009 and 2018, he was the Manager of the Vienna approach control unit. In 2013 he received his master degree in Business Administration with a specialization on Air Traffic Management from the Danube University of Krems (Austria).

Dietrich Klakow studied Physics from 1987 until 1991 at the Universities of Erlangen (Germany) and York (UK). He completed his PhD at the University of Erlangen in 1994. Until 1996 he was with Weizmann-Institute in Israel and then changed to the area of speech and language research, joined the Philips research lab. He held a lecturer position at Aachen University since 1999. Since May 2003 Prof. Klakow is professor at Saarland University in Saarbrücken (Germany), where he builds up a new research team which deals with algorithms for the human machine interaction. Other research focus is statistical natural language processing in particular question answering. In 2011 he received a Google Research Award.

Matthias Kleinert received his master degree in Computer Science from the Technical University of Braunschweig (Germany) in 2016. He joined DLR's Institute of Flight Guidance in 2012. He is a junior research scientist in the department of Controller Assistance since 2017.

Petr Motlicek is a senior research scientist at Idiap Research Institute. He received a master in electrical engineering and PhD in computer science at University of Technology, Brno, Czech Republic in 1999 and 2003, respectively. Between 2000 and 2001, he was a researcher at Oregon Health and Science University, Portland, USA, working on distributed speech recognition. Since 2012, he has been an external lecturer in the EPFL Electrical Engineering Doctoral (EDEE) program. Dr. Motlicek has largely contributed to well-known Kaldi open source ASR code.

Jürgen Rataj is head of the "Controller Assistance" department at the Institute of Flight Guidance. He has 30 years of experience with assistance systems in different application areas, e.g. aircraft engines, fixed wing aircraft, helicopter, road traffic as well as controller assistance. In addition, Mr. Rataj is assistant professor for ATM at the Ostfalia University of Applied Sciences.