Building Blocks of Assistant Based Speech Recognition for Air Traffic Management Applications

Matthias Kleinert[1], Hartmut Helmke[1], Heiko Ehr[1], Christian Kern[2], Dietrich Klakow[3], Petr Motlicek[4], Mittul Singh[3], Gerald Siol[1]
[1] Institute of Flight Guidance, German Aerospace Center (DLR), Braunschweig Germany,
[2] Austro Control, Vienna, Austria
[3] Spoken Language Systems Group (LSV), Saarland University (UdS), Saarbrücken, Germany
[4] Idiap Research Institute, Martigny, Switzerland
firstname.lastname@{dlr.de, austrocontrol.at, lsv.uni-saarland.de, idiap.ch}

***Abstract*—In air traffic control rooms around the world paper flight strips are replaced through different digital solutions. This enables other systems to access the instructed air traffic controller (ATCo) commands and use them for other purposes. Digital flight strip solutions, however, require manual input from the ATCo and, therefore, increase the workload. Recently the AcListant® project has validated that Assistant Based Speech Recognition (ABSR, which integrates a speech recognizer with an assistant system) could be a solution to avoid this increase of workload. However, adaptation of ABSR to new environments usually requires a lot of data, time and expertise, which makes the process expensive. The MALORCA project used machine learning (ML) algorithms to provide a generic, cheap and effective approach for adaptation. Therefore, ABSR was divided into conceptual modules that contain generic parts (building blocks) and domain specific models. As first show case ABSR was automatically adapted with radar data and voice recordings from Prague and Vienna approach. The fully trained system reaches command recognition rates (RR) of 92% (Prague) resp. 83% (Vienna) and command recognition error rates (ER) of 0.6% (Prague) resp. 3.2% (Vienna). The building blocks and models and their effect on RR and ER are presented in this paper.**

Keywords- Machine Learning, Assistant Based Speech Recognition, Building Blocks, Automatic Speech Recognition

I. INTRODUCTION

Problem

Recently, the Active Listening Assistant (AcListant®) project [1] has shown that a new type of Automatic Speech Recognition (ASR) [2] called Assistant Based Speech Recognition (ABSR) developed by Saarland University (UdS) and DLR [3]-[6] could be a solution to bring ASR applications from training facilities into the ATC operation rooms. The ABSR system listens to the controller-pilot communication and extracts the ATCo commands. The extracted commands are directly shown to the ATCo in the aircraft radar label or an electronic flight strip system. This gives the ATCo more free cognitive resources for other tasks, because he does not have to input all those commands manually. Instead the ATCo just has to correct the system in case of a false recognition. In simulation runs for the Dusseldorf approach area recognition rates better than 95% and error rates below 2% have been achieved.

In general ABSR integrates ASR with an Assistant System to provide a situational context based on radar data, weather information and additional data that is generated by the Assistant System. The provided context allows ABSR to modify/filter the results of ASR and to generate a search space for ASR that fits to the current air traffic situation. In the AcListant® project most of the time and budget (1.3M) have been used to adapt the necessary ABSR models and data to the needs of the ATCos controlling the Dusseldorf approach area. If ABSR needs to be transferred to other airports or ATCo positions (e.g. tower, departure) the models and data have to be changed manually for the selected environment. This process requires large amounts of data to be collected and expert knowledge for the necessary adaptations. In order to transfer ABSR to many different airports and ATCo positons this approach is too expensive and time consuming.

Solution

The Horizon 2020 SESAR project MALORCA (Machine Learning of Speech Recognition Models for Controller Assistance) [7] choose machine learning (ML) as a potential solution to provide a generic, cheap and effective way for the adaptation process. The solution makes use of the radar data and voice recordings that are generated every day in operation rooms around the world. This data is used as input for machine learning algorithms that automatically learn and adapt the necessary components of the ABSR system. For this purpose the system was divided into several conceptual modules with specific tasks. Each of the modules consists of different building blocks, models and data. The building blocks are active processes that fulfil certain tasks inside the modules. They are generic so that they can be reused for different controller positions and environments. Only the data and models are specific for each environment and need to be adapted or recreated through machine learning algorithms.

Paper Structure

In the next section we present related work with respect to machine learning and speech recognition applications in ATM. Section III describes the essential building blocks of an ABSR system, which conceptual model they belong to and how the modules interact with each other. In section IV we show the

The work is partly funded by SESAR Joint Undertaking (Grant Number 698824).

effect that each individual building block has on the overall result of ABSR. Before we come to the conclusions in section VI, section V describes the adaptations that are necessary to transfer ABSR to new environments.

II. Related Work

A. Speech Recognition Applications in ATM

Artificial intelligence (AI) and in particular machine learning (ML) applications have made a significant progress in the last few years, enabling computers to make a series of major breakthroughs that were previously impossible [8]. The AI winter ended 10 years earlier with the paper by Hinton [9] on pre-training neural networks (NN). One of the successful "application" fields of machine learning is automatic speech recognition (ASR), which has shown remarkable improvements in understanding human conversational speech. Speech recognition has developed for a very long time independently of the developments in the machine learning community. Especially the Work by Mikolov et al. [10] in the area of language modeling and Seide et al. [11] on acoustic modelling have boosted the interest in neural networks in the speech community. However, neural networks are very *data hungry* and thus difficult to apply to the ATM domain.

B. Supervised and unsupervised learning

In machine learning different kind of learning are considered: supervised learning, where labeled training data is available, unsupervised learning, where no labeled training data is available and semi-supervised learning, where some labeled data and a large amount of unlabeled data is available. A speech recognizer can either be treated as one big ML problem or it can be broken down into an ML problem for the so-called acoustic model and a separate ML problem for the so-called language model. For the acoustic model it is known that techniques like MLLR (Maximum Likelihood Linear Regression) or MAP (Maximum a Posteriori) can be used in a semi-supervised setting with some success [12]. Recently semi-supervised learning of neural network based acoustic models for special domains like YouTube videos has been performed [13]. For language models, neither unsupervised learning nor semi-supervised learning has been very successful. Only supervised learning is an established technique. For adapting non-neural LMs Bellegarda [14] gives a good overview. For adapting NN in a supervised fashion, this can either be done in a rescoring step [15] or by mapping the NN to a tree directly in the first pass for speech recognition [16].

III. Building Blocks of Assistant Based Speech Recognition (ABSR)

Figure 1 shows a rough overview of the four conceptual modules of an ABSR system, which are referred as DATA, TEXT, COMMAND and USER: The DATA module supplies the whole system with two types of input data: (i) dynamic and (ii) static. (i) Dynamic data is represented by the voice input signal and input/output of an assistant system (i.e. radar data, flight plan information, weather information, sequence data etc.) (ii) Static data for a given environment is represented by names of waypoints, runways, used frequency values etc. The TEXT module employs some of the data provided by the DATA module and executes Automatic Speech Recognition (ASR) related tasks on a given voice signal. This includes a speech-to-text (S2T) conversion, i.e. the speech signal is transformed via feature extraction into a sequence of words. Different word sequence hypotheses may result from the same voice input.

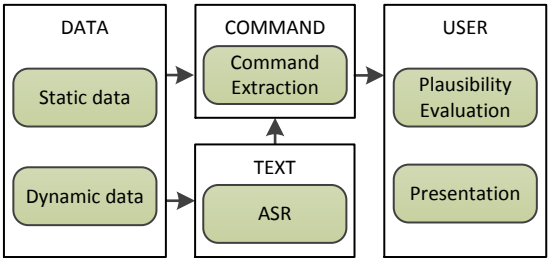

Figure 1 Conceptual overview of ABSR modules

The COMMAND conceptual module uses information provided by the DATA module to convert the different word sequence hypotheses generated by the TEXT module into air traffic controller (ATCo) command hypotheses. The output may still not be unique, i.e. different command hypotheses could result from the same voice input. Finally, the USER conceptual module selects a unique output, which is adequately presented to the controller. For this task command plausibilities and command predictions (i.e. possible ATCo commands) which are both provided by the COMMAND module. If the output after this process is not unique or all of the command hypotheses are not plausible, no output is shown to the ATCo (except of maybe just a callsign highlighting). The output of the TEXT module is finally rejected. The following example clarifies these ideas: We assume that the controller pronounces "turkish five kilo juliet maintain two two zero knots or greater descent three thousand feet".

The output generated by the TEXT module can be as follows

1. "turkish four kilo juliet maintain two two zero knots or greater descent eight thousand feet" and
2. "turkish five kilo juliet descent two two zero descent three thousand feet"
3. "hello four kilo juliet maintain two two zero or greater descent three thousand aeh"

The output generated by the COMMAND module can be as follows:

1. THY4KJ MAINTAIN SPEED 220 kt OR_GREATER, THY4KJ DESCEND 8000 ft and
2. THY5KJ DESCEND 220 none, THY5KJ DESCEND 3000 ft and

3. 4KJ MAINTRAIN 220 none OR_GREATER, 4KJ DESCEND 3000 none

Depending on the available context knowledge (i.e. the set of predicted commands) this could be corrected to:

1. THY5KJ MAINTAIN SPEED 220 kt OR_GREATER, THY5KJ DESCEND 8000 ft (THY4KJ not in the air)
2. THY5KJ DESCEND 220 none, THY5KJ DESCEND 3000 ft and
3. THY4KJ MAINTAIN 220 none OR_GREATER, THY4KJ DESCEND 3000 none (4KJ is an abbreviation)

The USER module with the plausibility evaluation (checker) should exclude the first (i.e. descent to 8000 feet is not expected) and the second (i.e. two descents do not (or seldom) occur in the same utterance) hypothesis. The third hypothesis is finally shown to the controller. Depending on the available Human Machine Interface, the ATCo will not even recognize the difference to the correct command transcription "THY4KJ MAINTAIN 220 kt OR_GREATER, THY4KJ DESCEND 3000 ft" with the correctly recognized units.

A. Elements of DATA conceptual module

The DATA conceptual module provides and generates the necessary information for the TEXT and COMMAND module of the ABSR System. As shown in Figure 2 it consists of the following building blocks that provide dynamic information:

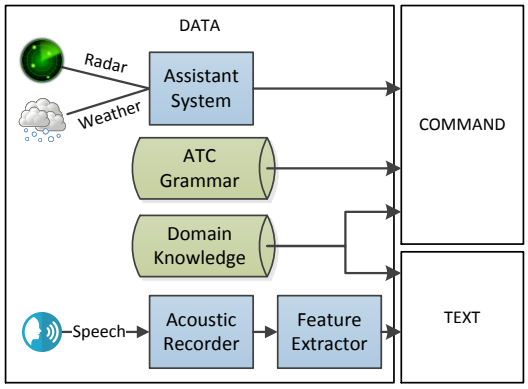

Figure 2 Components and relations of DATA module

- **Acoustic Recorder**, which records an analog speech signal of the controller's utterance into digital form. We use 8 kHz sampling rate.
- **Assistant System,** which provides e.g. radar data, flight plan information, weather data and additional information (e.g. landing sequence).
- **Feature Extractor,** which transforms the analog speech signal recorded from the Acoustic Recorder into set of conventional acoustic feature vectors X (i.e. mel-frequency cepstral coefficients).

In addition to the building blocks the DATA module in Figure 2 also contains the following static information:

- **ATC grammar**, i.e. a set of rules which describes how the different controller commands are combined by spoken words. Rules contain terminals (in lower case letters) and non-terminals (starting with upper case letter). The rules are based on ICAO phraseology [17]and controllers' local interpretations of the ICAO rules. A REDUCE command can be described by the grammar rules shown in Figure 3
- **Domain knowledge** is based on a given environment. This includes the runway names, handover frequency values, waypoint names and coordinates, pronunciations etc.

REDUCE := Callsign Type Value Qualifier Condition
Type := "reduce", "reduce to" …
Value := SpValue ["knots"]
Qualifier := "or greater" | "or less" | or *empty*
SpValue := "two one zero" | "two ten" | "two zero zero" | "two hundred", | "one nine zero" …
Callsign := …
Condition := …

Figure 3 Excerpt of Grammar for a REDUCE command

B. Elements of TEXT conceptual module

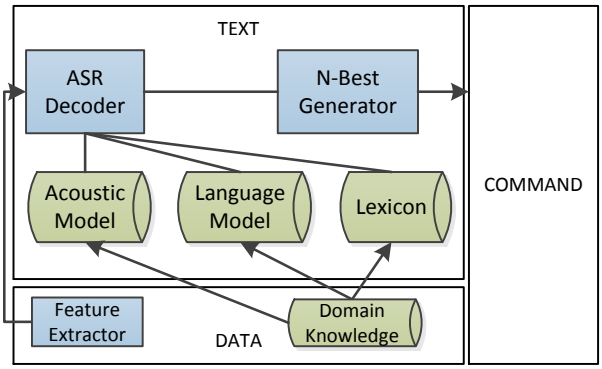

Figure 4 Components and relations of TEXT module

The TEXT conceptual module uses information provided from or extracted by the DATA module and sends its results to the COMMAND module. Its building blocks are shown in Figure 4:

- **ASR decoder:** The set of acoustic feature vectors X extracted by the Feature Extractor of the DATA module is transformed 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).
- **N-Best Generator**: Instead of extracting only the most probable sequences of words, the N-Best-Generator selects the N (e.g. 5) most probable word sequences W using the ASR decoder.

Besides the hitherto described building blocks, the TEXT module also contains three domain specific models that are necessary for ASR:

- The **lexicon** is one of the essential blocks of an automatic speech recognition system. To be able to model all possible commands spoken by ATCos, we expand standard CMU-Sphinx dictionary (of Carnegie Mellon University) [18] by all ATM in-domain words (such as airline names, waypoint names, etc. for a given approach area) to form an extended pronunciation lexicon.
- The **acoustic model** (AM) models the regional difference of speaking English (e.g. Czech English, or German English). Speaker independent versions and speaker dependent versions are possible. Usually, Deep Neural Networks (DNN) are used for acoustic modelling.
- **Language Model** (LM): Controllers often deviate from standard phraseology and hence, ATC suggested Context-Free Grammars (CFG) are too strict to learn the phraseology used by controllers. N-gram statistical language models can deal with deviations which cannot be easily modelled in CFG. An extended dataset is used for model training, further adapted to the ATC domain Language model [21]. A grammar-induced class based Language Model (LM) is used to generate sample sentences that fit the ATC grammar. These sentences are then combined with ground-truth transcripts available from training data and a 3-gram statistical LM is finally built, which is then used for Finite-State-Transducer-based ASR decoder.

C. *Elements of COMMAND conceptual module*

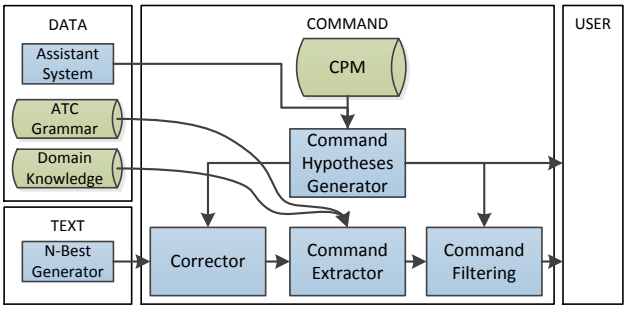

Figure 5 Components and relations of COMMAND module

COMMAND conceptual module is used to convert the raw text sequences obtained from the N-Best Generator of the TEXT conceptual module to ATC commands. Important building blocks used in the COMMAND module (shown in Figure 5 are:

- **Command Hypotheses Generator** to generate a set of commands, which are plausible (with respect to assistant system information of the DATA module) in the current air traffic situation.
- **Corrector** modifies the recognized word sequences from the N-Best Generator of the TEXT module by leveraging an e 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. on radar).
- **Command Extractor** transforms the corrected sequence of words (e.g. "good morning speedbird beta one charly reduce two ten or less") to ATC commands (e.g. "BWAB1C REDUCE 220 none OR_LESS"; none specifies that unit knots was not spoken).
- As the Command Extractor transform the (ambiguous) word sequences of N-Best-Filtering and its correction by Corrector to command sequences the output of Command Extractor might end up with multiple possible command sequences.
- The **Command Filtering** block selects the most plausible command sequence 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 also contains a **Command Prediction Model** (CPM). CPM contains the rules to generate the set of possible commands for target ATC approach. Details to CPM and its training by machine learning can be found in [19] and [20]

D. *Elements of USER conceptual module*

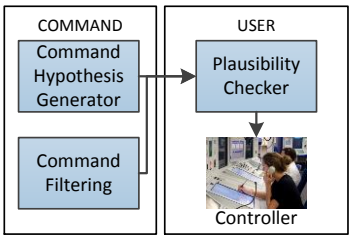

Figure 6 Components and relations of USER module

USER conceptual module (Figure 6) selects the most plausible ATC command from the set of outputs provided by the Command Filtering block to be finally shown to the user. More specifically, recognized commands, which either (i) do not fit to static domain knowledge (unknown waypoints), (ii) are not plausible (not predicted), (iii) are contradictory (climb and descend in same utterance for the same callsign), or (iv) have low recognition plausibility (low plausibility values from TEXT block) are not shown to the user (ATCo). This is the task of the **Plausibility Checker**. Assuming the output of the Command Filtering is "THY4KJ MAINTAIN SPEED 220 kt, THY4KJ DESCEND 8000 ft" the output to the ATCo could

- include both commands
- include only MAINTAIN SPEED resp. DESCEND command,
- include no command, resulting in NO_CALLSIGN NO_CONCEPT
- include only the callsign (THY4KJ NO_CONECT)

Even a correction to THY4KJ CONTINUE PRESENT_SPEED" is possible, if 220 knots is the current indicated airspeed (IAS) of THY4KJ.

IV. Effects of the Building Blocks

In this section, we measure the effects of different building blocks with respect to the final result, i.e. the recognized command(s) which are sent to the HMI of the ATCo. 18 hours of untranscribed training data for Prague and 18 hours for Vienna airspace were used as development data to adapt all building blocks to target ATC approach. Details with respect to learning the models are already provided in subsection III.B and in [19] and [21].

A. Scenarios

In order to evaluate the effects of the different building blocks, real radar and speech recordings from the Vienna and Prague airspace were used. For both airspaces, two different controller positions were taken into account.

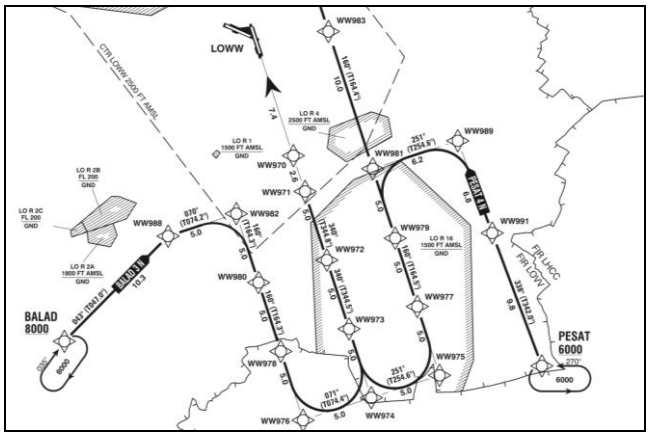

Figure 7 Arrival Transitions to Vienna Airport´s Runway 34

Table 1 shows the used voice recordings in more detail. The first row ("ATCo") contains the number of different ATCos separated into male and female. Next row shows the number of individual speech segments containing one complete voice transmission from an ATCo to a pilot. The segments are divided into manually transcribed transmissions, i.e. that the content of the transmission is known, and untranscribed transmissions. The last row shows the total amount of pure speech in minutes (speech without silence) that is contained in the speech segments.

Table 1: Scenario Voice Data

Airspace		Prague	Vienna
ATCo	Male	9	40
	Female	3	5
Speech Segments	Transcribed	3'039	3'012
	Untranscribed	12'034	14'361
Pure Speech in min	Transcribed	281	230
	Untranscribed	1'099	1'078

For Vienna the speech and the radar data were recorded for the Feeder (Final Director) position and for the BALAD sector controller when runway 34 was in use. It is the duty of sector "BALAD" to bring arriving traffic in an orderly flow on the so-called BALAD3N-Arrival-Transition, shown in Figure 7. This transition leads all arriving flights into a left-hand traffic pattern. The Feeder has to mix it with other arrivals coming in from the right-hand traffic pattern and to establish all arrivals on final approach track providing prescribed distances in between. For Prague equivalent positions have been chosen when runway 24 was in use.

B. Baseline with all building blocks

In [19] and [20] different metrics are defined to determine the quality of an ABSR system. Some of those key metrics are reused here measuring the influence of the different building blocks:

- Total number of given commands (**#TgC**),
- Command recognition rate (**RR**): number of correctly recognized commands, which are not rejected by CPM, divided by #TgC (a command is correct if both the callsign and the command type and the command value are correctly recognized) ,
- Command recognition error rate (**ER**): number of recognized commands which were not spoken and not rejected, divided by #**TgC**. These recognized commands are wrongly shown to the ATCo.

As baseline, we evaluated those metrics for a fully active ABSR System i.e. all the described building blocks are used. Those results are already reported in [19]. They are repeated here (see Table 2) and used to further describe the influence of the different building blocks. Since all building blocks are used, these results are also the best possible results for the current system with the available test and training data.

Table 2: Metrics When using all Building Blocks (Baseline)

Area	Total		Sector		Feeder	
in [%]	**RR**	**ER**	**RR**	**ER**	**RR**	**ER**
Prague	92.1	0.60	93.0	0.60	89.7	0.60
Vienna	83.3	3.21	80.8	3.42	89.3	2.71

Results based on 4,211 given commands from 3.84 hours of speech (no silence) for Vienna and 5,339 given commands from 4.69 hours for Prague. Due to rejection rate RR and ER do not sum up to 100%.

C. Effect of the Plausibility Checker

Table 3 shows the effect when the building block "Plausibility Checker" from the USER module is not active: All outputs provided by the "Command Filtering" of module COMMAND are accepted and shown to the ATCo. This also includes commands that might be unlikely or even contradictory.

Table 3: Metrics When Checker is not Used

Area	Total		Sector		Feeder	
in [%]	**RR**	**ER**	**RR**	**ER**	**RR**	**ER**
Prague	94.0	1.83	94.8	1.51	92.0	2.61
Vienna	84.9	6.21	82.7	7.31	90.0	3.53
Prague Delta	1.9	1.23	1.8	0.91	2.4	2.00
Vienna Delta	1.6	2.99	2.0	3.89	0.7	0.88

Rows with "Delta" show the absolute difference with respect to baseline with all building blocks. Improvements with respect to baseline are shown in green font, Degradations are marked with red font.

As shown in Table 3, the command recognition rate (RR) slightly increases, because with the "Plausibility Checker" even

correct recognitions are sometimes accidently discarded Compared to RR the error rate (ER), however, increases significantly, if the "Plausibility Checker" is not active. The total ER increases by a factor of 3 for Prague resp. 2 for Vienna.

D. Effect of Command Hypotheses Generator

Table 4 shows the effect when no command predictions are generated by the building block "Command Hypotheses Generator" from the COMMAND module. This has an effect on several other modules. In this situation, the "Corrector" (COMMAND Module) cannot modify ASR output by taking into account the current airspace situation. Also the "Command Filtering" block (COMMAND Module) is not able to base its command selection on the set of predicted commands. At last the "Plausibility Checker" (USER Module) cannot employ command predictions as well to filter out false recognitions.

TABLE 4: METRICS WHEN NO CONTEXT IS PROVIDED FROM HYPOTHESES GENERATOR

Area	Total		Sector		Feeder	
in [%]	RR	ER	RR	ER	RR	ER
Prague	87.5	6.66	89.5	5.52	82.8	9.47
Vienna	71.5	15.00	69.0	16.49	77.5	11.37
Prague Delta	-4.5	6.07	-3.6	4.92	-6.8	8.86
Vienna Delta	-11.8	11.78	-11.8	13.07	-11.8	8.66

Rows with "Delta" show the absolute difference with respect to baseline with all building blocks. Improvements with respect to baseline are shown in green font, Degradations are marked with red font.

As shown in Table 4 there is a clearly noticeable decrease in RR for Prague (-4.5%) and Vienna (-11.8%) if command predictions are available, but even more importantly there is a noticeable increase in ER. The total ER for Prague increases by a factor of 10 resp. almost 5 for Vienna. Command Hypotheses Prediction is fast, but it increases run time of TEXT module by a factor of 2.

E. Effect of (Speaker Dependent) Acoustic Model

The test and training data for Prague and Vienna contain speech recordings from different ATCos. In the baseline ABSR system, an individual speaker dependent acoustic model is created (i.e. each controller gets another acoustic model trained from his/her recordings). In order to evaluate the influence of this speaker dependent acoustic modeling on the overall results, we executed the same experiments with a speaker-independent acoustic model (i.e. the acoustic model is created from all speech recordings available from all ATCos).

Table 5 shows the results with a speaker-independent acoustic model. For Prague, there is a significant degradation in RR and ER compared to the baseline. The Vienna data instead shows a smaller degradation when the generic model is used. The reason for the bigger impact on the Prague results could be that the Prague speaker dependent acoustic models are based on four times more training data. We have roughly the same amount of training data, but in Prague from 12 controllers and in Vienna for 45 controllers. On the other hand the generic Vienna model is better, because it covers a wider amount of different voices.

TABLE 5: GENERATING AVERAGE ACOUSTIC MODEL FROM ALL ATCOS (SPEAKER-INDEPENDENT ACOUSTIC MODEL EMPLOYED IN ABSR)

Area	Total		Sector		Feeder	
in [%]	RR	ER	RR	ER	RR	ER
Prague	88.6	1.07	89.2	0.93	86.9	1.43
Vienna	82.6	3.42	79.6	3.65	89.8	2.88
Prague Delta	-3.5	0.48	-3.8	0.33	-2.7	0.82
Vienna Delta	-0.7	0.21	-1.1	0.23	-0.5	0.17

Rows with "Delta" show the absolute difference with respect to baseline with all building blocks. Improvements with respect to baseline are shown in green font, Degradations are marked with red font.

We repeated the same experiment with an acoustic model trained for a male resp. female controller. Table 6 shows the results when a model trained for one randomly selected (male) ATCo is used for all controllers. For Prague the RR and ER show a significant degradation compared to the baseline. The main reason is that the selected male model is delivering good results for most of the male controllers, but significantly worse results for the female controllers. For Vienna data the degradation is noticeable, but the selected male model here seems to fit quite well for male and female controllers.

TABLE 6: GENERATING ACOUSTIC MODEL FROM ONE MALE ATCO (I.E. GENDER-DEPENDENT ACOUSTIC MODEL).

Area	Total		Sector		Feeder	
in [%]	RR	ER	RR	ER	RR	ER
Prague	70.8	2.08	69.0	2.19	75.0	1.83
Vienna	82.2	3.53	79.8	3.75	87.9	3.00
Prague Delta	-21.3	1.48	-24.0	1.59	-14.7	1.22
Vienna Delta	-1.1	0.32	-0.9	0.33	-1.4	0.29

Rows with "Delta" show the absolute difference with respect to baseline with all building blocks. Improvements with respect to baseline are shown in green font, Degradations are marked with red font.

Table 7 shows the results when a model trained for one female ATCo is used for all controllers. The female models selected for this evaluation show significant deltas for both Vienna and Prague data in RR and ER.

TABLE 7: GENERATING ACOUSTIC MODEL FROM ONE FEMALE ATCO (GENEDER-DEPENDENT ACOUSTIC MODEL).

Area	Total		Sector		Feeder	
in [%]	RR	ER	RR	ER	RR	ER
Prague	76.3	2.09	78.9	2.13	70.1	1.98
Vienna	78.0	5.30	74.9	5.11	85.6	5.75
Prague Delta	-15.7	1.49	-14.2	1.53	-19.6	1.37
Vienna Delta	-5.2	2.08	-5.9	1.69	-3.7	3.04

Rows with "Delta" show the absolute difference with respect to baseline with all building blocks. Improvements with respect to baseline are shown in green font, Degradations are marked with red font.

F. Effect of N-Best Generator

The value N of the building block "N-Best Generator" from the TEXT module is set to 5 in the baseline (i.e. N-Best Generator delivers five most probable sequences of words which are then sent to the COMMAND module). To see the influence of this block we choose the following values for N: 1, 5 (baseline), 10, 20 and 50. The results are shown in Table 8.

If we analyze the results for N=1 for Vienna and Prague the ER is slightly better than the baseline, but the decrease in RR at

the same time has a bigger impact on the overall result. Changing N-Best to 10 slightly improves the RR for Prague and Vienna. The changes in ER are almost 0% and therefore have no real impact on the overall performance of the system. With N-Best values of 20 or 50 the performance (Comparison of changes in RR and ER) of the system still slightly improves for Vienna, but for Prague the performance almost stays the same. Increasing N has always a negative effect on the recognition time. From N=1 to N=5 it increases by a factor of 1.3. However, Table 8 also shows that bigger N values can also decrease performance.

TABLE 8: VARYING THE N-BEST OUTPUT OF ABSR

N-best	Area	Total		Sector		Feeder	
	in [%]	RR	ER	RR	ER	RR	ER
1	Prague	91.4	0.56	92.2	0.58	89.3	0.52
	Vienna	80.8	2.85	78.8	3.01	85.7	2.45
5	Prague	92.1	0.60	93.0	0.60	89.7	0.60
	Vienna	83.3	3.21	80.8	3.42	89.3	2.71
10	Prague	92.2	0.57	93.1	0.60	90.0	0.52
	Vienna	83.7	3.21	81.3	3.42	89.4	2.71
20	Prague	92.1	0.73	93.1	0.63	89.7	0.99
	Vienna	83.7	3.22	81.4	3.43	89.4	2.71
50	Prague	92.0	0.62	92.9	0.63	89.7	0.61
	Vienna	83.9	3.16	81.6	3.44	89.6	2.46

Improvements with respect to baseline (N-best 5) are shown in green font, Degradations are marked with red font.

V. ADAPTATION TO NEW ENVIRONMENTS

In the previous section, we have seen the effects of the different building blocks. In this section, we describe what is needed to adapt the described ABSR architecture to a new environment. This includes on the one hand the transformation to new approach areas, e.g. from Prague to Frankfurt, and on the other hand also addresses the challenge to transfer from approach area to tower or enroute environment. The conceptual modules (DATA, TEXT, COMMAND and USER) distinguish between buildings blocks and data/models. In most cases an adaptation of the building blocks is not necessary, but the used data/models have to be changed/trained to fit to the needs of the target environment.

In the following we draft the necessary adaptations to each individual conceptual module that are needed to transfer an ABSR system to another environment.

A. Adaptation of DATA conceptual module

A lot of the data provided and generated by the DATA Module is targeted to a specific environment. Obviously this data has to be changed for other target areas. This concerns the following parts:

- **ATC Grammar**: This set of rules includes, besides the standard ICAO phraseology, the local deviations controllers tend to make in different environments. A completely recreation is, therefore, not necessary, but parts of it have to be adapted manually to fit the needs of a specific target area.
- **Domain knowledge** consists of data that is mainly unique to a given environment (e.g. runways, waypoint names, used frequencies etc.). That means that this knowledge has to be defined manually again for every single environment.
- **Assistant System** in ABSR is used to provide information that is already available (e.g. radar data, weather information, sequence planning etc.) to other parts of ABSR. Some sort of assistant system that has these information is usually already being used at places were ABSR is useful. Of course those systems are not generic and interfaces to get the necessary data from those systems have to be implemented.

B. Adaptation of TEXT conceptual module

- **Acoustic model:** The acoustic model is automatically trained from transcribed or untranscribed data (see details from the MALORCA project [19]). If enough data is available for different controller and it is clear which controller is speaking a speaker-dependent model slightly improves recognition rates (see section IV.E)
- **Lexicon:** Each environment has its own waypoints and some local words for greetings and good-bye. These need to be manually added to the lexicon together with its possible pronunciations.
- **Language model:** As described in sect. II.B only supervised learning is an established technique for learning the local deviations of the language model. If only low amounts of labeled ATC text transcriptions are available the grammar itself can be used to label available text even from automatic transcriptions to ATC concepts. To further help with sparsity, large amounts of text from an existing ATC language models can be sampled in order to obtain a good coverage of possible unseen events, see [20], [21] for further details.

C. Adaptation of COMMAND conceptual module

For the COMMAND module adaptation is only necessary for the command prediction model (CPM), but very crucial since the information of the CPM is used by several other building blocks. The CPM of course is unique for a given environment and has to be recreated from scratch. As the MALORCA project has shown the CPM can be learned automatically from controller audio recordings resp. automatic command recognitions based on those recordings and corresponding radar data [19], [20].

D. Adaptation of USER conceptual module

No core parts of the ABSR system have to be modified for the USER conceptual module. Obviously most environments where ABSR can be used, already have their own HMI. The HMI itself of course is a decisive part, because it has a big influence on the end-user (controller) acceptance. So adaptations to an existing HMI probably have to be made to show the outputs of the ABSR system. A standardization concerning an

interface to HMIs and the format of the transmitted commands [22] might help, but that is not in the scope of this paper.

VI. CONCLUSIONS

ASR is used in ATC at least since the late 90s with different success. Recently, DLR and Saarland University have shown in the AcListant® project that acceptable command recognition rates are possible with ABSR which combines the output of an assistant system in form of context information with an ASR system. Adaptation of ABSR to different approach areas and controller positions, however, is expensive with respect to time and personal resources.

Based on the MALORCA project this paper presented a solution by dividing an ABSR system into different building blocks and models. The building blocks are reusable and the models can be automatically trained by unsupervised learning. Not all building blocks must be implemented, of course resulting in a performance decrease. Our experiments for Prague and Vienna show that the usage of the Command Hypotheses Predictor improves command recognition rate by a factor of 1.17 and reduces error rate by a factor of 11.3. A speaker dependent acoustic model can improve command recognition rate by a factor of 1.04 and reduce the error rate by 1.8. Generating the N best word sequences can slightly improve recognition rate (factor 1.04), but decrease error rate (factor 1.3).

Unsupervised learning of models is not new for ASR applications. MALORCA's invention is to use the plausibility checker of ABSR to subdivide automatic transcriptions into good and bad learning data. Using command prediction with plausibility checking can attenuate the problem of today's Artificial Intelligence systems that they have nondeterministic respectively difficult to predict behavior, i.e.using a second input sensor (radar data plus voice utterances) could enable certification of AI system even in ATM applications.

ACKNOWLEDGMENT

We would like to thank all the controllers who anonymously provided us with real world command examples and also our MALORCA partners from Austro Control and from Air Navigation Service Provider of Czech Republic.

REFERENCES

[1] The project AcListant® (Active Listening Assistant) http://www.aclistant.de/wp, n.d.

[2] T. Shore, F. Faubel, H. Helmke, and D. Klakow, "Knowledge-based word lattice rescoring in a dynamic context," Interspeech 2012, Sep. 2012, Portland, Oregon.

[3] H. Helmke, H. Ehr, M. Kleinert, F. Faubel, and D. Klakow, "Increased acceptance of controller assistance by automatic speech recognition," in 10th USA/Europe Air Traffic Management Research and Development Seminar (ATM2013), Chicago, IL, USA, 2013.

[4] 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.

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

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

[7] The project MALORCA (Machine Learning of Speech Recognition Models for Controller Assistance) http://www.malorca-project.de, n.d.

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

[9] G.E. Hinton and R.R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks. Science", Science, Vol. 313, Issue 5786, pp.504-507, 2006.

[10] 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

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

[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] ICAO: Manual of technical provisions for the aeronautical telecommunications network (ATN), DOC9705, 2nd eds, 1999, online available at https://www.icao.int/safety/acp/repository/_%20Doc9705_ed2_1999.pdf

[18] The CMU (Carnegie Mellon University) pronouncing dictionary http://www.speech.cs.cmu.edu/cgi-bin/cmudict , n.d.

[19] 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.

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

[21] A. Srinivasamurthy, P. Motlicek, M. Singh, Y. Oualil, M. Kleinert, H. Ehr and H. Helmke, "Iterative Learning of Speech Recognition Models for Air Traffic Control," in INTERSPEECH 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, Sep. 2018.

[22] 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