QUANTIFYING THE BENEFITS OF SPEECH RECOGNITION FOR AN AIR TRAFFIC MANAGEMENT APPLICATION

Hartmut Helmke[1], Youssef Oualil [2] Marc Schulder [2]

[1] German Aerospace Center (DLR), Institute of Flight Guidance, Braunschweig,
[2]Saarland University, Speech and Signal Processing, Saarbrücken
hartmut.helmke@dlr.de, youalil@lsv.uni-saarland.de, mschulder@coli.uni-saarland.de

Abstract: The project AcListant® (Active Listening Assistant), which uses automatic speech recognition to recognize the commands in air traffic controller to pilot communication, has achieved command recognition rates above 95%. These high rates were obtained with an Assistance-Based Speech Recognition (ABSR). An Arrival Manager (AMAN) cannot exactly predict the next actions of a controller, but it knows which commands are plausible in the current situation and which not. Therefore, the AMAN generates a set of possible commands every 20 seconds, which serves as context information for the speech recognizer.

Different validation trials have been performed with controllers from Düsseldorf, Frankfurt, Munich, Prague and Vienna in DLR's air traffic simulator in Braunschweig from 2014 to 2015. Decision makers of air navigation providers (ANSPs) are primary not interested in high recognition rates, respectively, low error rates. They are interested in reducing costs and efforts. Therefore, the validation trials, that were performed at the end of 2015, aimed at quantifying the benefits of using speech recognition with respect to both efficiency and controller workload. The paper describes the experiments performed to show that with ABSR support controller workload for radar label maintenance could be reduced by a factor of three and that ABSR enables fuel savings of 50 to 65 liters per flight.

1 Introduction

Recently, the venture capital funded project AcListant® has achieved command error rates below 1.7 %, resulting in command recognition rates above 95% [1]. The application domain was air traffic controllers' communication to pilots. Controllers do not require that each word of their utterance e.g., "good morning lufthansa three two bravo descend now altitude four correction three thousand feet" is recognized. However, they want that the relevant concepts are correctly extracted, i.e., in the example "DLH32B" for the name of the aircraft, "DESCEND" as the command type and "3000" as the target value of the command. The whole command, which must be correctly recognized, is "DLH32B DESCEND 3000". This requires a reliable speech recognition system. Our approach, Assistance-Based Speech Recognition (ABSR), jointly developed by DLR and Saarland University, uses speech recognition embedded in a controller assistant system, which provides a dynamic minimized world model to the speech recognizer. The speech recognizer and the assistant system improve each other. The latter significantly reduces the search space of the first one, resulting in low command recognition error rates.

Decision makers of airlines and ANSPs are not even interested in high command recognition rates. They are more interested in dollars and euros. Therefore, DLR performed end of 2015 validation trials with eight controllers from Austro Control and German air navigation provider DFS in Braunschweig. The aim was to quantify the benefits of using speech recognition with respect to both efficiency and controller workload [2].

In section 2 we present the background of our research, in sect. 0 we present ABSR, i.e., the integration of an assistant system to improve recognition rate of an ASR system (automat-

ic speech recognition). Then we present the experimental setup for quantifying benefits of ABSR. Section 5 presents the results before the conclusions follow.

2 Background

First integrations of ASR in air traffic control (ATC) systems especially for training started in the late 80s [3]. Nowadays enhanced ASR systems are used in ATC training simulators to replace expensive pseudo pilots. Although ASR systems are widely used in normal life (e.g., Siri®, telephone dialog systems, and interface for car navigation systems) and ATC phraseology is standardized, recognition and understanding controller pilot communication is still a big challenge and not solved satisfactory. Reasons are ATC vocabulary and syntax, as well as the variety of accents, speakers, and communication channels with different characteristics and especially controllers' needs to deviate from standard phraseology [5]. Cordero et al. (2012) reported of word detection rates not above 20% with different Commercial-off-the-shelf (COTS) recognizers [6]!

One promising approach to improve ASR performance is using context knowledge regarding expected utterances. These attempts go back to the 80s [7], [8]. This information may heavily reduce the search space and lead to less miss recognitions [4]. Our approach (Assistant Based Speech Recognition = ABSR) uses the output of an assistant system, i.e., DLR's AMAN 4D-CARMA [9], as context information. An "Assistant System" analyses the current situation of the airspace and predicts possible future states, the input of the "Hypotheses Generator" to predict the set of possible commands. This dramatically reduces the search space of the "Lattice Generator" [10], [11]. ABSR is detailed in the next section.

3 Assistance-based Automatic Speech Recognition

This section describes the standard ASR system and then shows how the context information, which is available and provided by the ATC assistance system (4D-CARMA in Figure 1), can be used to improve the recognition performance. This section also shows how plausibility values can be derived from the proposed speech recognition system as a confidence measure of its output. We use the public domain speech recognition engine KALDI [12], [13].

3.1 Automatic Speech Recognition

Automatic speech recognition is the task of translating spoken words captured by an audio signal into text also known as speech-to-text. Building an ASR system generally requires the development of two main components, namely, an acoustic model, which learns the statistical mapping between the audio signal, generally represented as overlapping feature vectors, and the phonemes or other linguistic units that form speech. The second component is the language model, which is responsible of modeling the word sequences and learning their probability of occurrence.

In AcListant®, the acoustic model was trained using the standard Hidden Markov Models (HMM) approach, which uses Gaussian Mixture Models as emission distributions, whereas we used a Context Free Grammar (CFG) as language model [10], which was developed based on the standard ATC phraseology manual [14] as well as common deviations observed in various pre-trials. We have also considered N-gram language models in our research [11].

3.2 Improving ASR using Context Information

The ATC assistance system bases its proposed command sequence on the state of a given airspace sector and its history. This state is primarily derived from radar information about the airspace as well as aviation domain knowledge. More precisely, the Hypotheses Generator in

Figure 1 creates with the output of the assistant system a set of commands which are plausible in the current airspace situation. The output represents the dynamic context, and is transformed into the search space for the ASR system. Dynamic context information is updated with 0.1 Hertz and typically contains a few hundred commands.

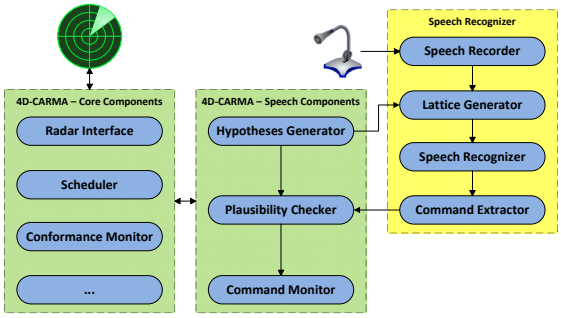

Figure 1 - Components of Assistant Based Speech Recognition [1], in green components of 4D-CARMA, in yellow components of core speech recognizer

When incorporating the context information into the ASR system, different factors are taken into account; in particular, the "run-time vs performance" compromise given that the resulting system should be able to perform online in a near real-time setting, i.e., the controller expects the recognized command within 3 seconds in at least 90% of the cases. We have investigated two different methods for integrating the context information into the speech recognition pipeline, namely, a pre-recognition approach [10], which uses the context information to automatically adapt the CFG to only cover utterances currently deemed possible. Then, this grammar is transformed into a Weighted Finite State Transducer (WFST), which is subsequently used to recreate the ASR search space. The second method is based on a Weighted Levenshtein Distance (WLD) [11], which extracts the ATC concepts from the recognized sentence in a first step, and then "corrects" these concepts based on the context information. The final hypotheses are extracted as the set of concepts that minimize the WLD. While the former approach uses the CFG, the latter uses a statistical N-gram language model. In order to achieve a good performance, these two approaches require high quality context information since they use the latter as a ground truth search space. The context error rate of the Hypotheses Generator is below 0.9% [2], i.e., only each hundredth command given by the controller was not expected.

3.3 Command Extraction

In order to effectively integrate the ASR output into the assistance system, an additional information extraction step is needed to extract the relevant ATC information. This task was successfully achieved by embedding XML-tags in the grammar itself. These tags are mapped to empty acoustic states and, therefore, can be integrated into an ASR system without affecting its performance. When the N-gram language model is used, however, the sequence labelling task was either achieved by transducing the raw ASR hypotheses using the WFST from the CFG or using a sequence tagger such as Conditional Random Fields (CRF). For instance, in this additional command extraction step, the ASR hypothesis "*air berlin three two bravo good morning turn left heading one two zero degrees*" is mapped to "*<callsign> air berlin three two bravo </callsign> good morning <command=turn_heading> turn <direction> left </direction> heading <degree> one two zero </degree> degrees </command>*". This command is subsequently returned to the assistance system as "*BER32B TURN_LEFT_HEADING 120*". The number of commands a controller gives in one single utterance to the pilot is not limited. Three to four often occur in high workload situations. The utterance "*austrian one two*

two alpha continue right turn heading three one zero descend three thousand feet cleared ils approach runway three four" results in the commands "*AUA122A TURN_RIGHT_HEADING 310*", "*AUA122A DESCEND 3000*" and "*AUA122A CLEARED_ILS 34*". The output of the "Command Extractor" is checked again by the "Plausibility Checker" in Figure 1 whether the recognized commands are reasonable in the current situation, e.g., do not produce conflicts. The "Command Monitor" analyzes the future behavior of the aircraft (radar data), whether they are in line with the "Command Extractor's" output.

3.4 Plausibility Values

ASR decoders can generally produce confidence scores, which reflect the certainty of the recognition for each word in the produced hypothesis. These scores are formalized as probabilities. In our research, the ASR system is based on the KALDI toolkit [13], and the ASR confidence scores were generated based on the Minimum Bayesian Risk (MBR) decoding approach [15]. In order to estimate the plausibility value for each ATC concept, the ASR confidence scores of the words that were tagged with the same label (of the corresponding ATC concept) were combined to form a concept-level confidence score.

4 Experiment Setup

The main purpose of the AcListant®-Strips project was to quantify the benefits of ABSR in ATC with respect to efficiency and controllers' workload. Air traffic controllers normally manage all aircraft information with flight strips. These strips contain static information about each flight such as call sign, weight category, destination, and route. Additionally, all clearances regarding altitude, speed, and course (heading or waypoints) are noted by the controller. Historically paper flight strips, maintained by pencil or ball pen, were in operation, but modern controller working positions use electronic flight strips or electronic aircraft labels. However, independent from this type, considerable controller effort is needed to manually maintain strip information consistent with commands given to the aircraft.

We evaluated two possible methods to insert given controller commands into the radar labels. The "manual" way is to completely use the mouse. The second input method is supported by an ABSR system. The controller may confirm or reject the output of the speech recognizer. In the latter case or if ABSR creates no output manual interaction of the controller is still necessary. Eight controllers from Germany and Austro Control (all native German speakers) participated in the trials. Two were female and six were male. They were between 22 and 53 years old (mean 36). They worked between 1 and 32 years as an active controller (mean 14 years). The participants neither were used to train the acoustic nor incorporated in the language model.

4.1 Compared Input Modalities

We evaluated two possible modalities to insert given controller commands into aircraft radar labels. The "manual" way is to use the mouse. By left clicking on one of the five interactive grey label cells (see left and middle part in Figure 2) a drop-down menu opens. The controller has to select the intended value (see right part in Figure 2), which are displayed in yellow afterwards. After completing the input of all necessary values for the respective aircraft, the controller has to confirm all these values by clicking on the green check mark (see middle part of Figure 2).

The second input modality is supported by ABSR, described in the previous sections. The voice channel between controller and pilot is analyzed by ABSR. The recognized commands, not the word sequences, are then visualized in yellow, i.e., they are still unconfirmed, in the corresponding five interactive cells shown in Figure 2. If the speech recognizer fails to cor-

rectly recognize the given command(s), the task of the controller was to manually input or correct the command as described above.

Figure 2 - Controller display for interaction with speech recognizer. Left: 5 interactive cells without contents; middle: speech recognizer output in yellow: right: drop down menu

4.2 Aircraft Scenarios

We used two different approach scenarios, i.e., aircraft sequences, of the Düsseldorf approach area. In the first scenario the controller is responsible for an area, which is normally controlled by two controllers, which are called the pickup and the feeder controller. Therefore, we call it the combined pickup/feeder (PF) scenario. On the other hand the controller only was responsible for the arrival traffic. The traffic demand was medium with approximately 35 arrivals per hour, see Figure 3). It lasts 60 minutes with a 5 minutes runway closure after 15 minutes and an emergency flight in the middle.

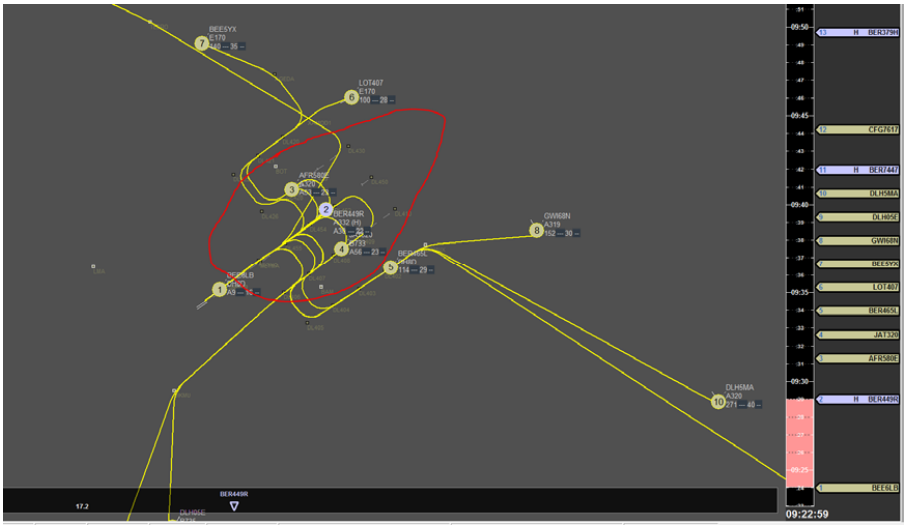

Figure 3 - Responsibility area of pickup/feeder scenario. On the right we see the planned sequence of the AMAN, in pink we see the planned runway closure

In the second scenario the controller only acts as feeder controller, but this time with very high traffic demand (60 arrivals per hour). The polygon in red in Figure 3 shows the responsible area of the feeder controller. The scenario lasts 45 minutes. 60 arrivals per hour on one runway are of course impossible due to minimum separation constraints. Holdings or long path stretchings are necessary to safely handle this amount of traffic. Controllers were asked to give feedback during pre-trials (performed in July and September 2015) when traffic demand would be too high. A simulated pickup controller would then automatically reduce the flow by deleting the next arrivals before they enter the control area of the controller. The controller, however, often forgot that feedback. In real life a deleting of aircraft is also not possible. Therefore, in the main trials, we automatically deleted the next two arrivals, if the final gets longer than 22 nautical miles (40.7 km). This eases also comparison of results.

5 Results

We calculate for each aircraft (except the first three landings) the difference between its flight time (from entering into scenario until touch down) and its earliest possible time predicted by AMAN 4D-CARMA. We exclude the first three aircraft because the beginning is always the "learning phase" of the controller. Table 1 lists the measurements of the average flight times.

Table 1 - Additional flight time in seconds

	Pickup / Feeder		Feeder	
Controller	ABSR+Mouse	Mouse	ABSR+Mouse	Mouse
A	121.6	190.1	285.1	266.6
B	111.1	163.6	375.0	401.6
C	187.8	215.4	286.6	291.3
D	147.9	150.7	316.8	344.8
E	305.3	271.9	437.6	347.4
F	208.6	529.1	376.2	340.3
G	141.8	345.3	250.7	360.2
H	178.0	156.4	250.4	364.7

Table 2 shows the corresponding mean, median and standard deviation of flight times. A paired t-tests falsifies the zero hypothesis, that mouse input modality is better than mouse plus speech recognition modality with an α (p-value) of 3.6%.

Table 2 - Results for average additional flight time, standard deviation and mean in seconds

Scenario	Input modality	Mean	Sigma = SD	Median
Pickup/Feeder	Mouse only	253	122	203
Pickup/Feeder	ABSR+Mouse	175	58	163
Feeder	Mouse only	340	40	346
Feeder	ABSR+Mouse	322	63	302

In the same way we calculated the average flown distance, the number of aircraft landed per hour and the number of given commands by voice to the pilots which were not entered correctly into the aircraft label. These were either forgotten to enter (often due to high work-load) or they were wrongly entered and not corrected. All these measurements confirmed the hypotheses that ABSR support for radar label maintenance in contrast to mouse only input results in better air traffic management (ATM) efficiency.

We also evaluated the influence of speech recognition on controller's workload. In detail we present here the time the controller needs for radar label maintenance with ABSR and without. We measured the duration of the time interval when the controller left clicks with the mouse on one of the five interactive label cells (Figure 2) and until clicking on the green check mark (ACCEPT). This roughly represents the duration needed to perform the label maintenance with the mouse. With ABSR support in most cases no clicking was necessary. Instead we used the Keystroke-Level-Model (KLM) [16] which defines execution times for different types of human-computer interaction, e.g. press or release a button, move the mouse to a specific position on the screen, the mental process of thinking what to do next.

Table 3 - Simulation time needed for command input in percentage

Scenario	Input modality	Mean	Sigma = SD	Median
Pickup/Feeder	Mouse only	30.6%	12.3%	28.3%
Pickup/Feeder	ABSR+Mouse	11.0%	3.1%	11.6%
Feeder	Mouse only	27.4%	11.2%	25.0%
Feeder	ABSR+Mouse	9.5%	2.0%	9.3%

We estimated the additional time compared to the "Mouse only" scenario with 1.2 seconds for every command that was accepted without any correction. This time correlates with the duration needed for a single mental process thinking of what to do next. Table 3 clearly

shows that without ABSR the controller uses nearly 30 percent of his time just for clicking. With ABSR support we reduce the time the controller needs for label maintenance by a factor of 3. These results are confirmed by further measurements (NASA TLX, ISA = Instantons Self-Assessment, secondary task). More details are provided in [2] and [17].

In subsection 3.4 we introduced the plausibility values. A high plausibility threshold results in lower command error rates, but on the other hand we have to accept also a lower command recognition rate. **Figure 4** shows the results if we change the plausibility value for the Munich controllers and the Vienna controllers. During the trials a plausibility value of 40% was used. Controllers from Munich (different from participants in trials) were included in the language and acoustic models, but no controllers from Vienna were included in the models. Vienna controllers, e.g. use the word "Servus" also as greeting like "Bonjour" or "Good morning". Munich controllers (and language model) use "Servus" only as "good bye".

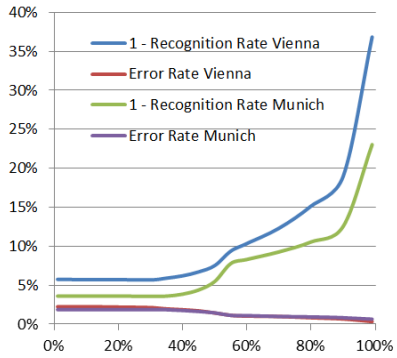

Figure 4 - Inverse recognition and error rate versus plausibility threshold

Concerning the error rate we observed no big difference between the German and the Austrian accent. If we look onto the (inverse) recognition rate, we, however, see a difference. The German accent or the German deviations from standard phraseology are modelled better in the ABSR system. There is room for improvements. We can choose any value between 0% and 40% for the plausibility value. It makes no significant difference. The ABSR system is robust. Plausibility values between 40% and 60% result in a decrease of the error rate. Of course the recognition rate decreases also, i.e., rejection rate increases. Plausibility values above 60% or even 70% have no added value, because error rate decreases only slightly, but recognition decreases dramatically.

6 Conclusions

This paper concludes our work with respect to ABSR. In 2011 Saarland University and DLR started with student works. In 2015, we demonstrated that command error rates below 1.7% are possible. We showed that ABSR improves the adaptation speed of an AMAN. This paper shows that ABSR significantly reduces controller workload and increases ATC efficiency. For the Düsseldorf approach area (after Frankfurt and Munich the biggest airport in Germany) we quantified the benefits to 50 to 65 liters less kerosene consumption per flight.

Acknowledgment

The work was conducted in the AcListant® project, which is supported by DLR Technology Marketing and Helmholtz Validation Fund.

References

[1] 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.

[2] H. HELMKE, O. OHNEISER, T. MÜHLHAUSEN, and M. WIES: *"Reducing Controller Workload with Automatic Speech Recognition."* in IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), Sacramento, California, USA, 2016.

[3] C. HAMEL, D. KOTICK, and M. LAYTON: *""Microcomputer System Integration for Air Control Training.","* in Special Report SR89-01, Naval Training Systems Center, Orlando, FL, USA, 1989.

[4] 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.

[5] SAID, M. GUILLEMETTE, J. GILLESPIE, C. COUCHMAN, and R. STILWELL: "*Pilots & Air Traffic Control Phraseology Study*." in International Air Transport Association, 2011.

[6] J.M. CORDERO, M. DORADO, and J.M. DE PABLO: "*Automated speech recognition in ATC environment*," in Proceedings of the 2nd International Conference on Application and Theory of Automation in Command and Control Systems (ATACCS'12). IRIT Press, Toulouse, France, 2012, pp. 46-53.

[7] 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.

[8] 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.

[9] H. HELMKE, R. HANN, M. UEBBING-RUMKE, D. MÜLLER, and D. WITTKOWSKI: "*Time-based arrival management for dual threshold operation and continuous descent approaches*." 8th USA/Europe ATM R&D Seminar, 29. Jun. - 2. Jul. 2009, Napa, California (USA), 2009.

[10] A. SCHMIDT: *"Integrating Situational Context Information into an Online ASR System for Air Traffic Control."* Master Thesis, Saarland University (UdS), 2014.

[11] 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.

[12] D. POVEY, A. GHOSHAL, G. BOULIANNE, L. BURGET, O. GLEMBEK, N. GOEL, ET AL.: "*The Kaldi Speech Recognition Toolkit*," in IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, Hilton Waikoloa Village, Big Island, Hawaii, US, Dec. 2011.

[13] KALDI: homepage: www.kaldi-asr.org, n.d.

[14] "*All clear phraseology manual*," in Eurocontrol, Brussels, Belgium, April 2011. Available: http://www.skybrary.aero/bookshelf/books/115.pdf

[15] V. GOEL AND W. J. BYRNE: "*Minimum bayes-risk automatic speech recognition*," in Computer Speech & Language, vol. 14, no. 2, pp. 115–135, 2000.

[16] D. KIERAS: "*Using the Keystroke-Level Model to Estimate Execution Time*," http://www-personal.umich.edu/~itm/688/KierasKLMTutorial 2001.pdf, 2001

[17] H. HELMKE, O. OHNEISER, M. WIES, AND M. KLEINERT: "*AcListant-Strips: Validation Results of Main Trials*," internal report, DLR-IB-FL-BS-2016-19, Braunschweig, 2016