CONTEXT-BASED RECOGNITION NETWORK ADAPTATION FOR IMPROVING ON-LINE ASR IN AIR TRAFFIC CONTROL

Anna Schmidt[†], *Youssef Oualil*[†], *Oliver Ohneiser*[‡], *Matthias Kleinert*[‡],
Marc Schulder[†], *Arif Khan*[†], *Hartmut Helmke*[‡], *Dietrich Klakow*[†]

[†] Spoken Language Systems, Saarland University (UdS), Campus, 66123 Saarbrücken, Germany
[‡] German Aerospace Center (DLR), Lilienthalplatz 7, 38108 Braunschweig, Germany

ABSTRACT

This paper presents an approach for incorporating situational context information into an on-line Automatic Speech Recognition (ASR) component of an Air Traffic Control (ATC) assistance system to improve recognition performance. Context information is treated as prior information to reduce the search space for recognition. It is integrated in the ASR pipeline by continually updating the recognition network. This is achieved by automatically adapting the underlying grammar whenever new situational knowledge becomes available. The context-dependent recognition network is then re-created and substituted for recognition based on these context-dependent grammars. As a result, the recognizer's search space is constantly being limited to that subset of hypotheses that are deemed plausible in the current situation. Since recognition and adaptation tasks can be easily performed by two separate parallel processes, on-line capabilities of the system are maintained, and response times do not increase as a result of context integration. Experiments conducted on about two hours of ATC data show a reduction in command error rate by a factor of three when context is used.

Index Terms— on-line ASR, situational context, air traffic control, assistance systems

1. INTRODUCTION

In this paper we describe how situational context information can be incorporated into on-line Automatic Speech Recognition to reduce error rates. We show this for speech recognition in the Air Traffic Control domain.

Air Traffic Controllers (ATCOs) are in charge of managing air traffic in a given airspace during a particular phase of travel, such as approach to an airport. Controllers are responsible for taking all relevant decisions concerning the current traffic situation and determine the aircraft sequence for landing. They issue commands to pilots steering aircraft, who then carry out the requested action.

Since the planning task ATCOs have to perform can – for example in case of dense traffic – become quite complex and mistakes can have safety-critical consequences, assistance systems are often employed to support ATCOs in their work. Such systems might for example suggest optimal aircraft landing sequences or future commands (*command advisories*) for the controller to issue to optimally manage the situation.

This work has in part been funded by Eureka project E! 7152, the Federal Republic of Germany through the Cluster of Excellence for Multimodal Computing and Interaction (MMCI), and DLR Technology Marketing and Helmholtz Validation Fund. UdS was responsible for ASR, DLR provided the arrival management system and evaluation data.

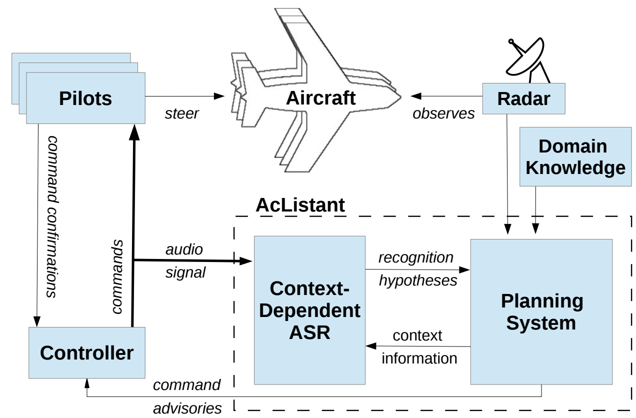

Fig. 1. Schematic view of flow of information in AcListant®.

In the AcListant® project an ATC assistance system for the landing (or *approach*) phase is being developed [1] [2] (see Figure 1). This assistance system generates specific command suggestions. Often controllers use these suggestions, but sometimes they have good reason to deviate from them, since an assistance system can not take into account all possible factors that might influence a situation during planning. Without any knowledge about the controllers actual decision, in such a situation the system would keep producing advisories based on incorrect assumptions about the current situation. These suggestions would be of no use to the controller, and it would take about 30 seconds for the planning component to be able to infer its mistake from radar data and adjust planning accordingly. This is the reason why in AcListant® an on-line speech recognition component is paired with the planning component generating command advisories. The ASR recognizes all commands the ATCO issues to pilots in the airspace and passes its recognition hypotheses to the planning system. By including ASR as an additional sensor, the planning system has access to feedback about the ATCOs actual decisions without having to wait for the consequences to become apparent in the radar.

In addition to generating command suggestions, the planning system can also provide the ASR component with information based on situational context. This information can be used to improve ASR performance. In order to not have faulty recognition hypotheses misguide the planning process, high ASR performance is crucial for the complete assistance system.

This paper is structured as follows: In Section 2 we give a short overview of related work. Section 3 then provides a short introduction of speech recognition using weighted finite state transducers. Then, in Section 4, we give background information about how the context-based information we utilize for ASR is generated. In Section 5 we discuss how this information is incorporated into the ASR pipeline to reduce error rates. In Section 6 we report and discuss experimental results for our system. Finally, Section 7 provides a conclusion and an outlook on future work.

2. RELATED WORK

In the past numerous works have incorporated different types of context information into ASR systems to improve performance. Among these are Young et al.'s works [8] and [9]. Making use of sets of contextual constraints of varying specificity their system generates several grammars. These grammars are then consecutively used during recognition of each utterance, starting with the most specific grammar and falling back to more unspecific ones until a satisfactory recognition hypothesis is found. This approach requires multiple decoding passes by design and can therefore be time consuming. Thus, it is not suitable for our application scenario, in which we depend on stable, short system response times. Still, we will adopt the general idea of grammar adaptation.

Fügen et al. also use dialogue context for ASR in their dialogue system proposed in [10]. As opposed to Young et al.'s system, in this approach the Recursive Transition Network representing the grammar is *continually* updated, which makes single-pass decoding possible. As will be discussed below, our integration approach goes in a similar direction.

In the work by Everitt et al. [11] a dialogue system is proposed that keeps record of a user's exercise routines. The ASR switches between specific pre-existing grammars tailored to different pieces of sports equipment when sensors indicate that the user is using a particular machine. In our system, however, context dependent recognition networks have to be dynamically generated.

The system presented in this paper directly builds on the work by Shore et al. [12]. The authors provide a proof of concept for our approach as they report first experimental results for using situational context information in ASR for the ATC domain. Reported results strongly indicate that incorporating situational context information significantly reduces recognition error rates. The authors evaluate using contextual information of varying degrees of specificity. The first type of context information Shore et al. test is information about the *callsigns* of the aircraft present in the airspace at the time of utterance. In aviation, the callsign is a unique identifier for aircraft typically consisting of airline and flight number. For example, the callsign *DLH12F* uniquely identifies the Lufthansa machine (*DLH*) that currently is assigned the flight number *12F* (*one two foxtrot*). In addition to callsign information, the more specific *speed* and *altitude* constraints were used. For this, inferred information about reasonable values for commands regarding *speed* and *altitude* of aircraft were utilized.

The authors incorporate context information by rescoring hypothesis lattices according to current situational context. Consequently, hypotheses not fitting the current situation receive a low probability. Reported results are promising, as SER decreased by about 18%. Since Shore et al.'s context integration approach makes a rescoring step necessary for every recognition and our system needs to function on-line, we opt for a different integration approach, as will be reported in Section 5.

3. ASR USING WEIGHTED FINITE STATE TRANSDUCERS

The main task in speech recognition is to uncover the string of words that corresponds to a given speech signal with the highest probability. Therefore, an ASR system typically takes an audio signal as input and outputs a a textual representation of its recognition hypothesis. More formally, it tries to uncover the most probable sequence of words (W) in the licensed language (L), given the observed audio signal (O) [3]:

$$\hat{W} = \arg\max_{W \in L} P(W|O) \tag{1}$$

In order to calculate $\hat{W}$, the equation is regularly reformulated in the following way [3]:

$$\hat{W} = \arg\max_{W \in L} \frac{P(O|W)P(W)}{P(O)} = \arg\max_{W \in L} P(O|W)P(W) \tag{2}$$

Our ASR system is built using the Weighted Finite State Transducer (WFST) approach introduced by Mohri in [7]. To implement it we used the Kaldi Speech Recognition Toolkit [4].

Generally speaking, WFSTs define relationships between sets of strings, and can be manipulated and combined using a number of existing standard operations [5]. In the WFST-based ASR approach, WFSTs can be used to represent parts of the decoding graph that is used for recognition, as well as the complete decoding graph itself.

In Kaldi, and therefore in our system, the decoding graph is constructed from the four basic components H, C, L, G all of which are WFSTs. Components H and C are generally concerned with the phone-level representation of speech. Transducer H translates 5-state Hidden Markov Models to Gaussian Mixture Models. Transducer C introduces phone context dependency by translating context independent phones into context dependent ones [5]. Component L represents the pronunciation lexicon in the decoding graph, as it translates phone sequences to words. Finally, component G contains a WFST representation of the grammar or language model (LM) used for recognition. In the AcListant® project, we are currently employing a grammar instead of an LM. G is the decoding graph component that is continually exchanged for a context-dependent version in our system.

In order to build the decoding graph for recognition, the H, C, L, and G components are combined. The equation for constructing the final decoding graph is [6]:

$$HCLG = det(H \circ min(det(C \circ min(det(L \circ G))))) \tag{3}$$

where $\circ$ denotes the *composition* operation that combines WFSTs [5]. For more details on WFST operations, WFST use in ASR and decoding graph construction in Kaldi please refer to [5], [7], and [6].

Once the decoding graph has been constructed it can be used for decoding. In our case this is done using a Kaldi implementation of Viterbi decoding (for details on Viterbi decoding see e.g. [3]).

4. CONTEXT INFORMATION

The context information we incorporate into the ASR pipeline is provided by the planning component in AcListant®. The planning system uses radar information about the airspace as well as aviation domain knowledge (cf. Figure 1) as a basis for generating command suggestions (so-called *command advisories*). Since it has access to these knowledge sources anyway, in addition to generating command advisories it can also use them to predict which commands

Fig. 2. System output for commands recognized by the system. The window on the left shows output for the baseline system version not using context information, while the window on the right shows output for version using context for the same input. Recognition errors are marked yellow, correct hypotheses are marked green.

are in the set of possible decisions a controller could take in the close future. This information is different from the set of command advisories, as the set of possible commands is much larger: It contains all commands the planning system deems *at all possible* to occur. In the remainder of this paper, this set of possible commands is referred to as *context information* for brevity. This is the information we utilize to limit the ASR search space.

Among of the main knowledge sources for generating context information are callsigns of the aircraft that are currently located in the airspace (cf. [12], [1]), information on aircraft position and heading (the direction of travel), and aircraft speed.

Once the planning system has computed the current context information, it passes it to the ASR component in an abstract form. For example, *DLH12F REDUCE 150 KT* would correspond to the command to aircraft *Lufthansa one two foxtrot* to *reduce speed* by *one hundred fifty knots*.

It is important to point out that one abstract command does correspond to many different possible utterances. Although the standard language in ATC is comparatively restricted English, and so-called *phraseologies* exist and should be adhered to (see for example [13] and [14]), controllers still introduce a lot of variation into their communication. Furthermore, non-nativeness is an issue that needs to be handeled when developing ASR for ATC.

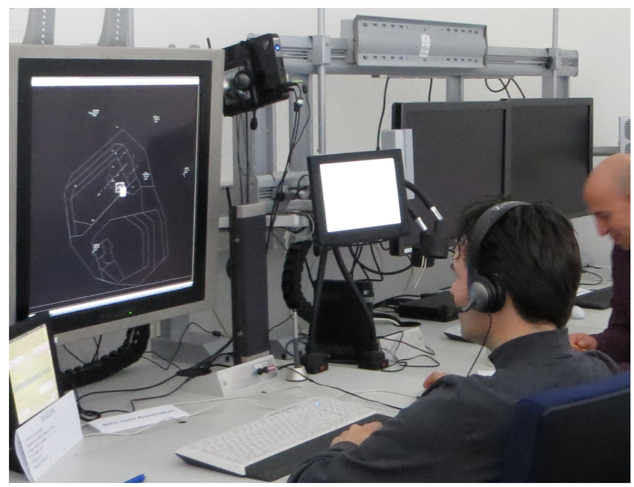

Fig. 3. Controller using assistance system during simulation for data collection.

5. INCORPORATING CONTEXT INTO ASR

When incorporating context information into the recognition process our two main objectives are to use it effectively to limit search space as much as possible to decrease the number of recognition errors, and at the same time not let this harm on-line capabilities of the ASR component.

In order to achieve this, we are using an approach that is based on the work of Shore et al. in terms of the type of context information we use, but in terms of *integrating* context information it is inspired by, for example, [10]. Our general approach is to use the context information available to automatically adapt the grammar used by the recognizer to only cover utterances currently deemed possible. Based on this new, restricted grammar we are then able to generate a new grammar WFST (G) that is smaller in size than the grammar WFST based on the basic, unrestricted grammar. We can then use this smaller WFST as a component during generation of a corresponding new recognition network for our Kaldi-based speech recognition system.

This integration approach has the advantage of enabling the system to have no processing delay caused by context usage: Since on-line recognition can continue running in one process while another parallel process prepares the new recognition network and then exchanges the networks, recognition is possible at all times with stable response time. Since the situation in the airspace only changes gradually, context information only changes gradually as well, which is why continuing to use the last existing recognition network while a new network is prepared does not pose a problem. Furthermore, context information is updated as soon as the planning system predicts new commands from changed radar data. Radar data is updated roughly every five seconds which means that intervals for context updates are just as short.

One major advantage of this strategy is that the ASR component only needs one decoding pass per recognition, since we are not employing hypothesis rescoring. This, of course, is beneficial for system response time. It should be mentioned that one possible downside to this approach is that hypotheses missing from context information are excluded as recognition hypotheses. However, as we will report below, the quality of context information is already high and improving further, so that the gain from the smaller search space is larger than the error introduced by incomplete context.

Recognition hypotheses are abbreviated to an abstract representation before transmitting them back to the planning system since the planner really only needs information about the ATC-relevant parts of the utterances. Greetings and other non-ATC-relevant utterance elements are not of importance to the planning system. However, information about the callsign addressed by the controller and the command issued to its pilot is crucial for carrying out the planning task.

6. RESULTS

Our experiments were run on data consisting of recordings of actual Air Traffic Controllers in simulations of real work situations. The simulations were performed in March 2014 at the German Aerospace Center in Braunschweig. The data consists of recordings of two controllers, one of them a native speaker of German and the other one a native speaker of Czech. Recordings were collected within simulation runs for the approach of Düsseldorf airport. The simulation ran over a total of eight hours in 13 simulation runs. Simulations are different from the simulated actual situation only in the sense that "pilots" are not located in actual aircraft during the simulation. For controllers the work performed during a simulation is the same as actual work. Figure 3 shows an air traffic controller during a simulation.

The sets of possible commands that are transmitted to the ASR component on average contained 239 predictions for the experiments reported here. They contain the actual controller decision as a prediction in about 96% of all cases.

We report recognition results separately for the two controllers recorded, since the acoustic model used has so far been trained only on German speakers of English with varying degrees of accents. The acoustic model has not been trained on any English with Czech accents. The acoustic model used in our evaluations has been trained on a total of six hours of ATC audio data recorded from various German controllers, and includes two hours of recordings of controller DE from a previous simulation.

For the German controller (DE), we recorded and evaluated 921 utterances, which amount to a total audio length of about 55 minutes. The average number of commands predicted in context information was 238 for these utterances. In general, for each utterance the context information that was available at the specific time of this particular utterance was used during evaluation. For the Czech controller (CZ) we recorded a total of 1007 utterances, which correspond to about 67 minutes of audio. The average size of context files for these utterances was 239 commands.

For evaluation, apart from the commonly used metric Word Error Rate (WER), ATC-specific evaluation metrics are used in the AcListant® project. These metrics are *Concept Error Rate* (ConER) and *Command Error Rate* (CmdER).

Concept Error Rate is a metric that it is restricted to the so-called *concepts* of an utterance. In the context of AcListant®, concepts consist either of the callsign information or of the remaining command elements. In order to calculate ConER, ATC-relevant utterance elements are automatically extracted from the recognition candidate. These segments are then translated to an abstract representation similar to the one that is used for representing context information. (See Figure 2 for illustration.) If any part of a concept is recognized incorrectly, this is enough for the complete concept to count as having been misrecognized. ConER is calculated similarly to WER, with the modification that instead of words, concepts are considered. For a better intuition, consider the example `DLH24F TURN_LEFT_HEADING_320`. This recognition hypothesis consists of two concepts, `DLH24F` and `TURN_LEFT_HEADING_320`. In case the controller really said *"Hello Lufthansa two four* ***four*** *turn left heading three two zero"*, ConER for this utterance would already be at 50%.

Command Error Rate is comparable to the commonly used Sentence Error Rate in the sense that is a binary measure: Either all relevant parts of an utterance are recognized correctly, or the recognition is assigned an error rate of 100%. CmdER is, however, restricted to the *concepts* of an utterance. In above example, CmdER would consequently be 100%. In case an utterance contains multiple commands, these are considered separately when calculating CmdER.

Both ConER and CmdER are suitable for evaluation in our application scenario, which ultimately uses ASR to provide feedback about the *ATC-relevant parts* of an utterance to the planning system. Recognition errors in non-ATC-relevant parts of utterances, e.g. greetings, are less critical for our application scenario. This is the reason we are interested in evaluating recognition errors occurring within these parts separately.

In both of the tables below, ***Dynamic fast*** refers to a system version that uses both callsign and command context information, but does not restrict callsigns to appear with particular commands. Thus, all callsigns listed in the context information can be combined with any command listed in the context information. ***Dynamic slow*** refers to a system version that allows each callsign to only occur with the specific commands it is listed with in context information. Therefore this system version is the more restrictive one of the two.

As a baseline we use our ASR system not enhanced by any situational context information (***Without Context***).

Controller DE	**Without Context**	**Dynamic Fast**	**Dynamic Slow**
%WER	7.32	**7.21**	8.04
%ConER	15.55	**9.68**	9.87
%CmdER	30.77	16.38	**15.63**
Respone time	0.83	2.72	7.17

Table 1. Evaluation results for recordings of German controller. Total length of audio in evaluation for German controller: 55 minutes

Controller CZ	**Without Context**	**Dynamic Fast**	**Dynamic Slow**
%WER	16.93	**10.71**	10.73
%ConER	35.94	10.60	**9.94**
%CmdER	68.88	18.80	**17.30**
Respone time	0.43	2.55	7.14

Table 2. Evaluation results for recordings of Czech controller. Total length of audio data in evaluation for Czech controller: 67 minutes

For both controllers, introducing context information into the recognition pipeline clearly improves recognition results over the baseline. We observe the largest improvements for the metrics most relevant for ATC: For ConER we improve performance by more than a factor of two for the German controller, and by more than a factor of three for the Czech controller. For our most important metric, CmdER, error rates are also reduced roughly by a factor of two for the German controller. For the Czech controller, CmdER is reduced by a factor of more than four.

A very interesting aspect of the reported results is the difference in error rates we observe for the two different controllers we tested, both regarding general ASR performance and error rate reduction through context information use. The difference in general performance has a rather obvious explanation: Our system has almost exclusively been developed and trained with data recorded from ATCOs who are German native speakers. Therefore, our acoustic model fits the German controller very well, since is has seen many different German accents for English, and even about two hours of data from this very speaker in a previous simulation. Furthermore, our grammar has been developed using the same data. Although English for ATC is a rather restricted variety of English, native language still plays a role for the optimal grammar. To give an indication, in our experiments we observed 95 out of vocabulary words (OOV) for the Czech controller, while we encountered only 63 OOV for the German controller. OOVs are observed whenever controllers deviate from standard ATC phraseology.

One aspect that is left to discuss is the fact that the improvement introduced by context information is less large for the German ATCO. The most important aspect for explaining this effect is that the *context error rate* (i.e., the average number of cases in which the correct hypothesis is missing from the context information) is much higher for the German controller than for the Czech controller. For the Czech controller data the correct recognition hypothesis was missing from context information in only 1.72% of all cases, while for the German controller it was missing in 4.92% of all cases. This means that the dynamic context for the German controller excludes correct hypotheses from the recognition network, causing the recognizer to fail. The quality of dynamic context is improved further in the AcListant® project, and therefore this issue is expected to have less of an impact very soon.

One very promising aspect of our experiments is that using context in ASR for air traffic control actually has the highest impact precisely in situations where the system is *not* tailored to the specific characteristics of the speaker. Our acoustic model has never seen data recorded from a person with a Czech accent. Our grammar is not optimized to Czech ATCO habits. Still, the improvements caused by contextual information are so large that in the end the best performing system manages to narrow the performance gap between the two controllers to a difference of just 1.67% in CmdER. The reason for this is that although the acoustic model is a not a good model for the Czech speaker, context information helps to solve ambiguities that occur when the acoustic scores of recognition candidates are *comparable*, by discarding hypotheses that are implausible in the current situation. Overall, our results therefore indicate that using contextual information makes speech recognition for ATC more robust against speaker variations.

Reported response times include the time spent on recognition network recreation *and* on recognition itself. Differences in response times between different system versions using context are caused by differences in the size of the resulting context-dependent recognition network component G. The size of this component influences the determinization step that is necessary during recognition network recreation, which is carried out in the background by a process running in parallel. The actual recognition on the newly created networks are comparable to the baseline response times. Therefore, on-line performance of the recognizer is preserved.

7. CONCLUSION AND FUTURE WORK

In this paper we showed that incorporating situational context information into on-line speech recognition for ATC significantly improves recognition results for the most relevant evaluation metrics. Our experiments showed an improvement in CmdER by a factor of up to three. For scenarios in which situational context only changes gradually, our integration method can achieve this error rate reduction without slowing down the actual recognition process, because recognition and recognition network recreation are run by two parallel processes.

Still, a trade-off exists between system response time and the specificity of context information to which we adapt the recognition network: Incorporating the most specific context information possible (*Dynamic Slow*) is currently only beneficial in cases where the quality of available context information is very high. However, the system version that is a little less restrictive (*Dynamic Fast* provides shorter response times and still shows considerable improvements over the non-context-enhanced system in situations where context quality is lower.

Furthermore, we showed that using context information makes the ASR robust against speaker variation. For an accent that was not present in the training data, we in fact observe the largest improvements. Using context manages to narrow the performance gap between the two different controllers tested to under 2%.

We expect the quality of dynamic information to become even better and more stable in the future, which will further increase the beneficial effect of context use.

In the future we are planning to combine the benefits of using contextual information with the use of confidence measures. Once the use of confidence measures is integrated into the ASR system, it will only return hypotheses about which it is sufficiently confident to the planning system. Possibly incorrect recognition hypotheses will be held back in order to avoid negatively influencing the planning system's performance. With this modification, the overall performance and usability of the assistance system should increase even further in the future.

Since using statistical language models instead of a grammar-based approach is currently being tested in the AcListant® project, another task to be addressed in future work will be to investigate context adaptation for language models.

8. REFERENCES

[1] Oliver Ohneiser, Hartmut Helmke, Heiko Ehr, Hejar Gürlük, Michael Hössl, Thorsten Mühlhausen, Youssef Oualil, Marc Schulder, Anna Schmidt, Arif Khan, and Dietrich Klakow, "Air traffic controller support by speech recognition," in *Proceedings of the 5th International Conference on Applied Human Factors and Ergonomics AHFE 2014*, T. Ahram, W. Karwowski, and T. Marek, Eds., Kraków, Poland, 19-23 July to appear: 2014, AHFE.

[2] Hartmut Helmke, Heiko Ehr, Matthias Kleinert, Friedrich Faubel, and Dietrich Klakow, "Increased acceptance of controller assistance by automatic speech recognition," in *Tenth*

USA/Europe Air Traffic Management Research and Development Seminar (ATM2013), June 2013, pp. 1–10.

[3] Daniel Jurafsky and James H. Martin, *Speech and Language Processing (2nd Edition) (Prentice Hall Series in Artificial Intelligence)*, Prentice Hall, 2 edition, 2008.

[4] Dan Povey and Arnab Ghoshal, "The kaldi speech recognition toolkit," in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding : ASRU 2011 proceedings*, Piscataway, NJ, 11-15 December 2011, IEEE, DK.

[5] Matthias Wölfel and John McDonough, *Distant Speech Recognition*, Wiley, Chichester, UK, 2009.

[6] Kaldi Online Documentation, "Description of decoding graph construction on kaldi homepage," http://kaldi.sourceforge.net/graph.html, March 2014.

[7] Mehryar Mohri, "Weighted finite-state transducer algorithms. An overview," in *Formal Languages and Applications*, Carlos Martín-Vide, Victor Mitrana, and Gheorghe Păun, Eds., vol. 148 of *Studies in Fuzziness and Soft Computing*, pp. 551–563. Springer Berlin Heidelberg, 2004.

[8] Sheryl R. Young, Wayne H. Ward, and Alexander G. Hauptmann, "Layering predictions: Flexible use of dialog expectation in speech recognition," in *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI89*. 1989, pp. 1543–1549, Morgan Kaufmann.

[9] Sheryl R. Young, Alexander G. Hauptmann, Wayne H. Ward, Edward T. Smith, and Philip Werner, "High level knowledge sources in usable speech recognition systems," *Commun. ACM*, vol. 32, no. 2, pp. 183–194, February 1989.

[10] Christian Fügen, Hartwig Holzapfel, and Alex Waibel, "Tight coupling of speech recognition and dialog management – dialog-context dependent grammar weighting for speech recognition," in *International Conference on Speech and Language Processing, ICSLP 2004*, Jeju Island, Korea, October 2004, ISCA.

[11] Katherine Everitt, Susumu Harada, Jeff A. Bilmes, and James A. Landay, "Disambiguating speech commands using physical context," in *Proceedings of the 9th International Conference on Multimodal Interfaces ICMI*. 2007, pp. 247–254, ACM.

[12] Todd Shore, Friedrich Faubel, Hartmut Helmke, and Dietrich Klakow, "Knowledge-basedword lattice rescoring in a dynamic context," in *InterSpeech 2012*, September 2012.

[13] Deutsche Flugsicherung GmbH (DFS), *Nachrichten für Luftfahrer: Bekanntmachungen über die Sprechfunkverfahren*, Langen, January 2012.

[14] International Civil Aviation Organization (ICAO), "Annex 10 to the convention on international civil aviation. Aeronautical Telecommunications. Communication procedures including those with pans status," Tech. Rep., International Civil Aviation Organization (ICAO), October 2001.