Real-Time Integration of Dynamic Context Information for Improving Automatic Speech Recognition

Youssef Oualil[1]*, Marc Schulder*[1]*, Hartmut Helmke*[2]*, Anna Schmidt*[1]*, Dietrich Klakow*[1]

[1]Spoken Language Systems Group (LSV), Saarland University, Saarbrücken, Germany
[2]German Aerospace Center (DLR), Institute of Flight Guidance, Braunschweig, Germany
{youssef.oualil,marc.schulder,anna.schmidt,dietrich.klakow}@lsv.uni-saarland.de
hartmut.helmke@dlr.de

Abstract

The use of prior situational/contextual knowledge about a given task can significantly improve Automatic Speech Recognition (ASR) performance. This is typically done through adaptation of acoustic or language models if data is available, or using knowledge-based rescoring. The main adaptation techniques, however, are either domain-specific, which makes them inadequate for other tasks, or static and offline, and therefore cannot deal with dynamic knowledge. To circumvent this problem, we propose a real-time system which dynamically integrates situational context into ASR. The context integration is done either post-recognition, in which case a weighted Levenshtein distance between the ASR hypotheses and the context information, based on the ASR confidence scores, is proposed to extract the most likely sequence of spoken words;, or pre-recognition, where the search space is adjusted to the new situational knowledge through adaptation of the finite state machine modeling the spoken language. Experiments conducted on 3 hours of Air Traffic Control (ATC) data achieved a reduction of the Command Error Rate (CmdER), which is used as evaluation metric in the ATC domain, by a factor of 4 compared to using no contextual knowledge.

Index Terms: speech recognition, situational context, Levenshtein distance

1. Introduction

Automatic speech recognition is a cornerstone of innovation in many technology applications due to the large demand for voice-enabled systems. Although ASR performance improved significantly over the last decade, it is far from being a solved problem, in particular for large vocabulary scenarios. Limited vocabulary (closed domain) applications, however, are deployed more successfully. Moreover, use of situational/contextual information about a task, generally referred to as context, can significantly improve performance [1]. Early usage of context goes back to Young et al.'s works [2, 3], who used sets of contextual constraints to generate several grammars for different contexts. Fügen et al. [4] used a dialogue-based context to update a Recursive Transition Network (RTN) to improve ASR quality of a dialogue system. Everitt et al. [5] proposed a dialogue system for gyms, which, based on the exercise routine, would switch its ASR component between pre-existing grammars tailored to different sports equipments.

This work has in part been funded by DLR Technology Marketing, the Helmholtz Validation Fund and Deutsche Forschungsgemein-schaft (DFG) under grant SFB 1102 (Sonderforschungsbereich).

The ATC domain is a prime example of heavy use of context. Air Traffic Controllers (ATCOs) manage a given airspace by issuing verbal commands to pilots for sequencing and maintaining aircraft separation. They rely on situational knowledge acquired through multiple modalities, including radar derived aircraft state vectors (comprising position, speed, altitude, etc.), flight plans and a history of previous commands.

The same information can also be used to improve ASR performance. Shore et al. [6] integrated context information in the ATC domain via lattice rescoring, whereas Schmidt et al. [7] proposed a dynamic context-based adaptation of the recognition network on the finite-state-machine level. The former was a limited proof of concept that took more than $30s$ per utterance. The latter requires $\approx 7s$ for an optimal integration of the context, referred to as dynamic-slow in [7]. Given that new context information becomes available with each radar update cycle every 5 seconds, this is still very slow. Moreover, both works are based on grammars that model the standard communication phraseology of the International Civil Aviation Organization (ICAO) [8]. However, analysis of ATC communication has shown that $\geq 25\%$ of issued commands do not follow this phraseology. These deviations vary among ATCOs, airports and countries, making it difficult to model in the grammar.

We propose a new approach based on N-gram Language Models (LM) to automatically capture these variations. The proposed approach performs the context integration post-recognition, using a modified weighted Levenshtein distance between the ASR hypothesis and the context information, using the ASR confidence scores as weights. This is in contrast to the previously proposed approach [7], which adjusts the search space to new situational knowledge through adaptation of the finite state machine that models the spoken language.

Proceeding, we introduce the ATC task as well as a domain-specific ASR system in Section 2. Section 3 details how to dynamically utilize context information to improve ASR performance and how it can be applied to the ATC domain. Section 4 evaluates the proposed approach in comparison to a standard ASR system and to [7]. Finally, we conclude in Section 5.

2. ASR in Air Traffic Control

2.1. AcListant® System

The task of air traffic control aims at maintaining the safe, orderly and expeditious flow of air traffic. ATCOs apply strict separation rules to direct aircraft safely and efficiently, both in their respective airspace sector and on the ground. Since controllers have an incredibly big responsibility and can face high work-

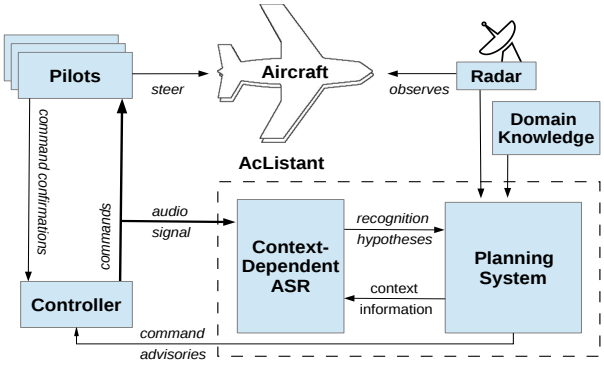

Figure 1: Schematic view of information flow in AcListant®.

loads in busy sectors, different planning systems have been proposed to assist them in managing the airspace. These assistance systems suggest e.g. an optimal sequence for the controller to implement via verbal radio communication with the aircraft pilots. These systems, however, do not know the controller's actual commands and thus react slowly to deviations of the controller from the system's plan or require the controller to enter the issued commands via mouse/keyboard, indirectly increasing the workload that they were designed to reduce. To alleviate this problem, we proposed an Active Listening Assistance (AcListant®) system [9, 10] which extends the planner to include a background ASR system, ideally replacing the mouse/keyboard feedback. Conversely, ASR can also benefit from the context information used by the assistant system [11] to improve its performance. Figure 1 shows the information flow in AcListant®.

2.2. Dynamic Context Information

Similar to ATCOs, an 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. The planning system forms a search space of all physically possible commands in the current airspace state from which to extract advisory commands, i.e. a sequence of commands to optimize a set of ATC criteria. This search space represents our **dynamic context**, and can be seen as a command-level search space for the ASR system. Typical dynamic context information generated by the AcListant® system contains a few hundred commands in the standardized ICAO phraseology format [8] (see example Table 1), comprising an aircraft callsign (e.g. DLH23B $\cong$ *Lufthansa two three bravo*) followed by a goal action to execute and a goal value to achieve (e.g. REDUCE 250 $\cong$ *reduce speed two five zero knots*). Section 3 will introduce how such context information can be used to improve ASR performance.

3. Dynamic Context-based ASR

This section shows how abstract context information can be successfully used to improve ASR performance in real-time. Without loss of generality, the proposed framework is illustrated

Callsign	Command Action	Value
DLH23B	REDUCE	250
AFR2A	TURN_RIGHT_HEADING	60
BER9000	RATE_OF_DESCENT	2000
KLM8739	DESCEND	100

Table 1: Excerpt from dynamic context information generated by AcListant®. It shows an ICAO abstraction of four different actions that can be issued by the controller to an aircraft.

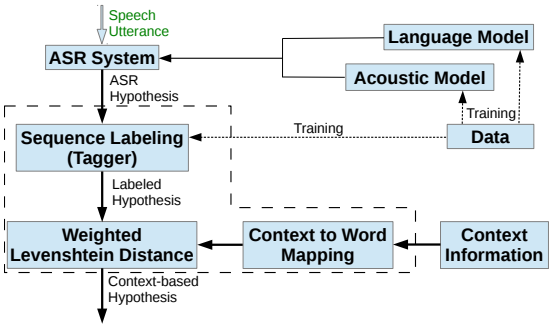

Figure 2: Diagram illustrating the proposed post-recognition integration of context information into an ASR system.

through examples from the ATC domain. Figure 2 shows the proposed system architecture for a post-recognition integration of the context. The following sections introduce its different components in more details.

3.1. From Grammar to N-gram LM

We have proposed in [7] to integrate context information into ASR by dynamically updating the recognition network of a grammar-based ASR. Context information is used to update the terminal values of the relevant grammar actions/rules. The updated grammar is converted to a finite-state machine (FSM) in the AT&T format [12], and a new recognition network is created accordingly. However, ATC data has shown that $\geq 25\%$ of the issued commands do not follow the standard ICAO phraseology [8]. These deviations vary between ATCOs, airports, etc., making it difficult to always create a suitable grammar, resulting in a strong decrease in ASR performance. As an alternative to this, we investigate the usage of N-gram language models as a straightforward way to automatically and cheaply capture these variations. In particular, a trigram LM is trained on a combination of data from ATC simulations and on synthetic data generated from the grammar.

3.2. Sequence Labeling for Semantic Concept Extraction

The main benefit of using Context-Free Grammars (CFG) for structured tasks such as ATC is their capability to automatically extract semantic concepts relevant to the task (e.g. callsign, command action and value, etc). This is 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.

While N-gram LMs are easy to train and, unlike handwritten grammars, do not require any manual inspection of data, they cannot automatically extract relevant semantic concepts. We propose to solve this problem by using sequence labeling. The labeller takes the raw ASR hypotheses as input and automatically **detects and extracts** the semantic concepts relevant to the target task. In the ATC domain for instance, the hypothesis "*air france two alpha hello turn right heading six zero degrees*" is mapped to "`<callsign>` *air france two alpha* `</callsign>` *hello* `<command=turn_heading>` *turn* `<direction>` *right* `</direction>` *heading* `<degree>` *six zero* `</degree>` *degrees* `</command>`". We have investigated two different sequence labellers, namely a Conditional Random Field (CRF)-based tagger [13] and a CFG-based token tagger similar to the one used in [6, 7]. Both systems achieved similar performance, extracting $\approx 90\%$ of semantic concepts.

3.3. Context-to-Word Mapping

Although context information can significantly improve ASR performance, its integration is not straightforward. This is mainly due to two reasons: 1) The information is partial and can only be used to modify the probability of a few words in the vocabulary, generally those with semantic relevance to the task. 2) The context is not necessarily represented as a fully realized natural language phrase. In the case of the ATC domain (see examples in Table 1), a mapping from context to all possible word sequences is necessary. For instance, the callsign `BER9000` contains the ICAO symbol `BER`, which can be spoken as *berlin*, *berlin air* or *air berlin*, and the flight number 9000, which, according to official phraseology, should be spoken as *nine zero zero zero*, but is often spoken as *nine triple zero* or *nine thousand*. The combination of these different realizations already leads to nine possible verbalizations that can be chosen by the controller to address this particular aircraft.

Our approach maps all abstracted callsigns and commands in a given context to their sentence realizations. The context-to-word mappings of callsigns are generated dynamically for the given context. Command actions (e.g. `REDUCE`, `DESCEND`, etc.) do not change over time and thus are captured directly by the language model. The resulting set of word sequences is used to improve ASR performance as explained in the next section.

3.4. Weighted Levenshtein Distance

Integration of domain knowledge into a language model is generally done through adaptation, which requires a sufficient amount of data to estimate reliable statistics. Adaptation approaches are usually offline techniques and can be slow, which makes them unsuitable for a real-time system. Capturing long range dependencies (e.g. a callsign can be seven words long) during adaptation requires long history N-grams, which in turn requires even larger amounts of data. Unfortunately, dynamic context information is unsuitable for providing such data, as it contains only crucial information for a specific task, but lacks knowledge of the possible ways in which it might be realized in natural language.

We propose to overcome this problem using a **Weighted Levenshtein Distance (WLD)**. We take the set of context word sequences (resulting from the context-to-word mapping) and treat it as the search space representing our ground truth, i.e. all commands the ATCO might speak in the current situation. Given the ASR hypothesis of a speech input, we calculate the WLD between that hypothesis and every context word sequence. The sequence that minimizes the WLD is extracted as the most likely spoken utterance. The rest of this section introduces the mathematical formulation of this approach.

Formally, standard Levenshtein Distance (LD) [14] between two sequences of words A and C is given by

$$LD(A,C) = \underset{p}{\operatorname{argmin}}\{\alpha \cdot s(p) + \beta \cdot i(p) + \gamma \cdot d(p)\} \quad (1)$$

where p is a sequence of operations from $\{sub, ins, del, none\}$ which change the sequence A into C, whereas s is the number of substitutions, i insertions and d deletions in p. α, β and γ are fixed weights (generally =1), or operation costs. In ASR, LD is typically used as an evaluation metric to calculate the edit distance between a hypothesis A and a given unique transcription C. LD assumes a uniform probability distribution for all words in A and C.

In this work, however, we are interested in choosing the most likely word sequence taken from a set of context information verbalizations (interpreted as potential ground truths), given an ASR hypothesis. In other words, our task is to extract the context hypothesis C closest to the ASR hypothesis A. In neither of these do the words have uniform weights. The ASR hypothesis contains confidence scores, which reflect the certainty of the recognition for each single word, formalized as probabilities. The context information provides probabilities for each callsign-command combination, which are applied to their respective verbalizations.

Let $\{w_a^j\}_j$ and $\{w_c^k\}_k$ be the probability weights associated with the words in A and C respectively, then WLD is given by

$$WLD(A,C) = \underset{p}{\operatorname{argmin}}\{\alpha \cdot S(p) + \beta \cdot I(p) + \gamma \cdot D(p)\} \quad (2)$$

where

$$S(p) = \sum_{s \in p} w_a^{j_s} \cdot (1 - w_c^{k_s}),\ I(p) = \sum_{i \in p} w_a^{j_i} \text{ and } D(p) = \sum_{d \in p} (1 - w_c^{k_d})$$

In our case, $w_a^{j_s} \cdot (1 - w_c^{k_s})$ is the cost of replacing an (ASR) word with a confidence $w_a^{j_s}$ by a (context) word of probability $w_c^{k_s}$. Similarly, the confidence score $w_a^{j_i}$ is used as the cost for the corresponding (ASR) word being inserted, whereas $1 - w_c^{k_d}$ is the cost of the (context) word being deleted. The integration of confidence scores aims at penalizing the inclusion of words where the system is less confident, as well as context hypotheses with a low probability of occurrence. Hence, the proposed approach does neither simply reduce the search space as done by [7] nor re-weight hypotheses for rescoring [6], but actively forces the system to choose the most likely hypothesis through a combination of ASR confidence and contextual likelihood of occurrence. This strategy relies on high quality context information. In the case of our ATC task, context accuracy is $\geq 99\%$.

3.5. WLD applied to ASR for ATC

Let $A = \{A_{cs}, \{A_{com}^{cs}\}_{com}\}$ be the semantics extracted from the ASR hypothesis using the sequence labeling system described in subsection 3.2. We assume that each hypothesis contains a single callsign A_{cs} in addition to one or multiple command goals to achieve $\{A_{com}^{cs}\}_{com}$.

The semantics considered here match the type of information that is dynamically updated in the context information (e.g. speed value, flight level value, etc). Similarly, let $\mathcal{C} = \cup_{cs}\{(C_{cs}, \{C_{com}^{cs}\}_{com})\}$ be the set of all possible context-based ground truths resulting from the context-to-word mapping of subsection 3.3. This set consists of all callsigns in the context and the command actions applicable to them. The context-based hypothesis is extracted according to

$$\begin{aligned} H &= \underset{C \in \mathcal{C}}{\operatorname{argmin}}\{WLD(A,C)\} \quad (3) \\ &= \underset{C \in \mathcal{C}}{\operatorname{argmin}}\{WLD(A_{cs}, C_{cs}) + \sum_{A_k \in \{A_{com}^{cs}\}} \underset{C_j \in \{C_{com}^{cs}\}}{\operatorname{argmin}} WLD(A_k, C_j)\} \end{aligned}$$

The resulting context-based hypothesis $H = \{H_{cs}, \{H_{com}^{cs}\}_{com}\}$ is then used to update the ASR hypothesis by "correcting" the misrecognized information in $\{A_{cs}, \{A_{com}^{cs}\}_{com}\}$. The extension of our approach to the N-best hypotheses case is straightforward by simply applying the same method for all N hypotheses separately, and then extracting the context-based hypothesis with the minimum WLD.

4. Experiments and Results

We evaluate the proposed approach using recordings of actual ATCOs running simulations of different scenarios for the approach of Düsseldorf airport. They were performed in October

Table 4: ASR results for the three controllers using different ASR systems with and without context information

ASR Systems	Czech Male ATCO				German Male ATCO				German Female ATCO			
	WER	ConER	CmdER	$\overline{\text{CmdER}}$	WER	ConER	CmdER	$\overline{\text{CmdER}}$	WER	ConER	CmdER	$\overline{\text{CmdER}}$
Grammar	7.89	12.28	22.16	19.13	3.49	8.99	16.74	13.11	6.46	10.34	18.13	15.23
Trigram LM	6.53	9.59	19.03	17.02	2.21	6.03	11.86	9.12	4.89	7.81	14.49	10.09
Grammar+Context	7.32	4.82	8.33	7.87	3.16	4.36	7.44	7.26	5.79	4.18	7.25	7.07
Trigram LM+Context	**6.07**	**2.91**	**5.08**	**3.39**	**2.08**	**3.42**	**6.02**	**4.83**	**4.78**	**2.96**	**5.39**	**3.47**

Controllers	German Male	German Female	Czech Male
Number of Commands	835	1323	960
Total Duration (min)	55	96	72
Avg. Time/Command (s)	3.95	4.35	4.54

Table 2: Recording statistics for the three controllers

	German Male		German Female		Czech Male	
Run-Time (s)	R_t	C_t	R_t	C_t	R_t	C_t
Grammar+Context	**2.05**	4.88	**1.54**	4.90	**1.61**	4.82
LM+Context	2.73	**0.23**	2.05	**0.22**	1.76	**0.22**

Table 3: Real-time performance: average recognition time (R_t) per command and context integration time (C_t), both in seconds

2014 at the German Aerospace Center in Braunschweig as part of validation trials for the AcListant® system. The aim was to show that its planning and ASR components improve each other when combined. The data consists of recordings of three controllers, a male German, a female German and a male Czech native speaker. All commands were issued in English. Each controller ran a total of 7 simulations for different scenarios. Table 2 presents recording statistics for the different controllers.

The dynamic context information is updated every 5 seconds by the assistant system [11]. It contains on average 359 possible commands, in contrast to the 239 used in [7]. This 50% increase improved the probability of containing the actual spoken commands in the context from 96% to 99%. ASR was performed using the KALDI software [15] and the ASR confidence scores were generated based on the Minimum Bayesian Risk (MBR) decoding approach [16]. The acoustic model is a GMM-based triphone model trained on 20 hours of ATC data. The data is a combination of the freely available Air Traffic Control Simulation Speech Corpus (ATCOSIM) [17] and previously collected data from past simulations. The combined corpus contains data from 21 controllers who are either native German, Swiss or French. 80% of them are male. The ATC-grammar used in these experiments is an extension of the one proposed in [7]. It implements the standard ATC phraseology [8] in addition to most common deviations observed in the training data. The LM, on the other hand, is a trigram model trained on a combination of the aforementioned ATC corpus and synthetic data generated from the grammar. For evaluation, in addition to the commonly used Word Error Rate (WER), the ATC-specific evaluation metrics *Concept Error Rate* (ConER) and *Command Error Rate* (CmdER) are used. ConER is restricted to the ATC-relevant semantic concepts of a given utterance, which are extracted using the sequence labeling approach (subsection 3.2). A concept can be either a callsign or a command, e.g. REDUCE_250. The CmdER metric requires the entire sequence of concepts to be correct. In the case where the sequence labeling system fails in extracting ATC-concepts, it returns NO_CALLSIGN or NO_COMMAND. These cases are counted as misrecognitions (deletions), even though they have no impact on the planning system since they do not provide any information. Therefore, we also report the CmdER after excluding these utterances (noted $\overline{\text{CmdER}}$) to estimate the misrecognition rate which negatively affects the planning system. When the ground truth contains no given command we have counted this in CmdER as an error as well.

Table 4 reports the ASR results for the three ATCOs using different ASR systems with and without context information. The approach Grammar+Context is the one proposed in [7].

The first conclusion we can draw from these results is that the trigram LM clearly outperforms the grammar-based system with and without context information, despite the grammar having been extended specifically for the training data to include new rules beyond the standard ATC phraseology. The trigram LM automatically captures these variations and assigns probabilities of occurrence reflecting their importance, contrary to the grammar which treats all rules and words as equally likely. We can also conclude from these results that context information strongly improves the ATC-related metrics (ConER, CmdER and $\overline{\text{CmdER}}$), whereas it only slightly improves the WER of either system. This is mainly due to the context information being dense and partial, i.e. it can only be used to improve the probability of occurrence of specific words in the vocabulary (the words which are ATC-relevant). These words, however, form only a small portion of the vocabulary (digits, letters, airlines, etc) leading to this slight improvement of the WER.

Table 4 also shows that the AcListant® system is robust and stable, i.e. the ConER and CmdER of the ASR+Context systems do not change much between speakers, despite not having been trained on a Czech accent and having seen only a small amount of female audio recordings. This robustness also covers variations in speed of issuing commands. This becomes clear when we look at Table 2, which shows that the male German controller issues commands a lot faster than the other controllers, while CmdER increases by only $\approx 1.4\%$.

Table 3 shows that the proposed approach integrates the context information a lot faster ($C_t = 0.2s$) than the grammar+context approach ($C_t = 4.8s$) by [7]. The latter is slow for a real-time system which receives new context information every 5 seconds. The recognition time, however, is within real-time range for both systems, $R_t = 1.74s$ and $R_t = 2.10s$, respectively, for utterances which are more than $3.5s$ long.

5. Conclusion

We have presented a novel approach to integrate dynamic context information into a speech recognition system in real-time. This approach extracts the ASR hypothesis in a first step and then uses a weighted Levenshtein distance to update this hypothesis in a context-based search space of all possible word sequences. We evaluated our approach in the controller pilot communication domain where command error rate is an important metric. Using dynamic context information we could reduce command error rate from 15.8% to 3.9% using three hours of speech data. In the future, we will investigate how performance is influenced by less reliable context information.

6. References

[1] G.-J. M. Kruijff, P. Lison, T. Benjamin, H. Jacobsson, H. Zender, and I. Kruijff-Korbayová, "Situated dialogue processing for human-robot interaction," in *Cognitive Systems*, ser. Cognitive Systems Monographs. Berlin/Heidelberg, Germany: Springer Verlag, 2010, vol. 8, ch. 8, pp. 311–364.

[2] 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. Detroit, MI, USA, August 1989*, 1989, pp. 1543–1549.

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

[4] C. Fügen, H. Holzapfel, and A. Waibel, "Tight coupling of speech recognition and dialog management - dialog-context dependent grammar weighting for speech recognition," in *INTERSPEECH 2004 - ICSLP, 8th International Conference on Spoken Language Processing, Jeju Island, Korea, October 4-8, 2004*.

[5] K. Everitt, S. Harada, J. A. Bilmes, and J. A. Landay, "Disambiguating speech commands using physical context," in *Proceedings of the 9th International Conference on Multimodal Interfaces, ICMI 2007, Nagoya, Aichi, Japan, November 12-15, 2007*, 2007, pp. 247–254.

[6] T. Shore, F. Faubel, H. Helmke, and D. Klakow, "Knowledge-based word lattice rescoring in a dynamic context," in *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*, 2012, pp. 1083–1086.

[7] A. Schmidt, Y. Oualil, O. Ohneiser, M. Kleinert, M. Schulder, A. Khan, and H. Helmke, "Context-based recognition network adaptation for improving on-line asr in air traffic control," in *2014 IEEE Spoken Language Technology Workshop (SLT 2014)*, 2014, pp. 2–6.

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

[9] H. Helmke, Y. Oualil, J. Rataj, T. Mühlhausen, O. Ohneiser, H. Ehr, M. Kleinert, and M. Schulder, "Assistant-based speech recognition for ATM applications," in *Proceedings of* 11^{th} *USA/Europe ATM R&D Seminar (ATM2015)*, Lisbon, Portugal, June 2015.

[10] DLR, "The AcListant® project." [Online]. Available: http://www.AcListant.de

[11] H. Helmke, R. Hann, M. Uebbing-Rumke, D. Müller, and D. Wittkowski, "Time-based arrival management for dual threshold operation and continuous descent approaches," in *Proceedings of* 8^{th} *USA/Europe ATM R&D Seminar (ATM2009)*, Napa, California, USA, June - July 2009.

[12] M. Mohri, F. C. N. Pereira, and M. Riley, "A rational design for a weighted finite-state transducer library." in *Workshop on Implementing Automata*, ser. Lecture Notes in Computer Science, vol. 1436. Springer, 1997, pp. 144–158.

[13] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01, San Francisco, CA, USA, 2001, pp. 282–289.

[14] S. Konstantinidis, "Computing the edit distance of a regular language," *Information and Computation*, vol. 205, no. 9, pp. 1307 – 1316, 2007.

[15] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011.

[16] V. Goel and W. J. Byrne, "Minimum bayes-risk automatic speech recognition," *Computer Speech & Language*, vol. 14, no. 2, pp. 115–135, 2000.

[17] K. Hofbauer, S. Petrik, and H. Hering, "The ATCOSIM corpus of non-prompted clean air traffic control speech," in *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may 2008.