ISCA Archive
http://www.isca-speech.org/archive

INTERSPEECH 2012
ISCA's 13th Annual Conference
Portland, OR, USA
September 9-13, 2012

Knowledge-Based Word Lattice Rescoring in a Dynamic Context

Todd Shore[1], *Friedrich Faubel*[1], *Hartmut Helmke*[2], *Dietrich Klakow*[1]

[1]Spoken Language Systems, Saarland University, Saarbrücken, Germany
[2]Institute for Flight Guidance, German Aerospace Center (DLR), Braunschweig, Germany
{todd.shore,friedrich.faubel}@lsv.uni-saarland.de

10.21437/Interspeech.2012-328

Abstract

Recent advances in automatic speech recognition (ASR) technology continue to be based heavily on data-driven methods, meaning that the full benefits of such research are often not enjoyed in domains for which there is little training data available. Moreover, tractability is often an issue with these methods when conditioning for long-distance dependencies, entailing that many higher-level knowledge sources such as situational knowledge cannot be easily utilized in classification. This paper describes an effort to circumvent this problem by using dynamic contextual knowledge to rescore ASR lattice output using a dynamic weighted constraint satisfaction function. With this method, it was possible to achieve a roughly 80% reduction in WER for ASR in the context of an air traffic control scenario.
Index Terms: lattice rescoring, knowledge-based, context-sensitivity

1. Introduction

Many domains are contextually very rich in that any utterance is highly dependent on the situation in which the dialog is situated [1]. An extreme example of such is air traffic control (ATC), in which controllers issue verbal commands to aircraft pilots in order to maintain aircraft separation and to assist in landing. The commands which a controller issues are based on the current state of the airspace as well as future plans regarding incoming aircraft, weather conditions, etc. This situational knowledge is acquired through multiple modalities, including radar-derived aircraft state vectors (comprising the position, speed, altitude, descent rate, reduce rate and heading), flight plans and the previous command history.

Just as contextual information is used in ATC for planning and issuing commands, this information can be utilized to improve the accuracy of automatic speech recognition (ASR) in the domain of ATC. One simple method would be to reduce the perplexity of the task by constraining the number of possible referents denoted by a given ATC command to those which would plausibly be issued the command given the current and previous state of the airspace as well as the information denoted by the command itself. Similarly, possible airspeed and altitude values could be restricted based on knowledge of the current situation. In a grammar-based language model, this can be achieved by reweighting rules in order to penalize them in a particular context [2]. This procedure, however, turns out to be unwieldy for a complex, context-dependent scenario, where the range of possible numeric values (e.g. aircraft airspeed) may change depending on the particular referent of the sentence (e.g. a particular aircraft). Hence, in this paper, we investigate the use of word lattice rescoring for this task. This is done in the domain of ATC, which is particularly well-suited to this form of context-based rescoring due to the relative regularity of the grammar used by controllers in issuing ATC commands: ATC commands are issued in a standardized subset of English which is formally specified by the International Civil Aviation Organization (ICAO) [3]. Thus, concepts can easily be extracted and then be evaluated using contextual knowledge.

The remaining part of the paper is organized as follows: Section 2 briefly describes the ATC task along with the grammar that we use to recognize commands; Section 3 introduces the concept of knowledge-based lattice rescoring and then explains how this form of rescoring can be applied to the task of ATC. Finally, Sections 4 and 5 present experiment results and a discussion.

This work has been supported by the Federal Republic of Germany, through the Cluster of Excellence for Multimodal Computing and Interaction (MMCI).

2. ATC Task Description & Grammar

The primary objective of an air traffic controller is to maintain the separation of aircraft in the airspace which is under control. This includes guiding approaching aircraft to their runway threshold, safely integrating departing aircraft into an aircraft stream passing through their sector as well as guiding passing aircraft through the airspace. This is primarily accomplished by issuing verbal radio commands to aircraft pilots so that the aircraft follow a pre-planned flight plan which takes many factors into account such as other aircraft, weather and the current capacity of the airspace.

2.1. Command format

ATC commands are issued using standardized ICAO phraseology [3], comprising a single aircraft callsign (e.g. *Air France four one eight* ≙ `AF418`) followed by a goal action to execute (e.g. *descend* ≙ DESCENT) and a goal value to achieve during that action (e.g. *flight level seven zero* ≙ `FL70`). Table 1 shows a subset of the ATC command goal types that have been used in this paper, including the corresponding value arguments.

Type	Values	Example
DESCENT	`ALT`	*descend altitude* `ALT` *feet*
DESCENT	`FL`	*descend flight level* `FL`
REDUCE	`SPD`	*reduce speed* `SPD` *knots*
TURN	`DIR`, `HDG`	*turn* `DIR` *heading* `HDG`

Table 1: Subset of the used ATC command goal types with their corresponding value arguments. Here, `ALT`, `FL`, `SPD`, `DIR` and `HDG` denote the altitude in feet and flight level (i.e. hundreds of feet – depending on the air pressure), speed, direction (left /right) and heading (angle in degrees), respectively.

2.2. Contextual information

The commands an ATC controller issues are dependent on the current and past state of the airspace in question; Knowledge about this state is acquired primarily through flight plans and radar-derived aircraft state vectors (comprising the position, speed, altitude, descent rate, reduce rate and heading). In order to make this process more efficient, controller assistant systems are being developed to support controllers in monitoring the airspace as well as in planning and implementing the future airspace state. Figure 1 shows a GUI of such a system at the example of an arrival manager (AMAN), which is used to assist controllers in managing aircraft separation and improving flight efficiency.

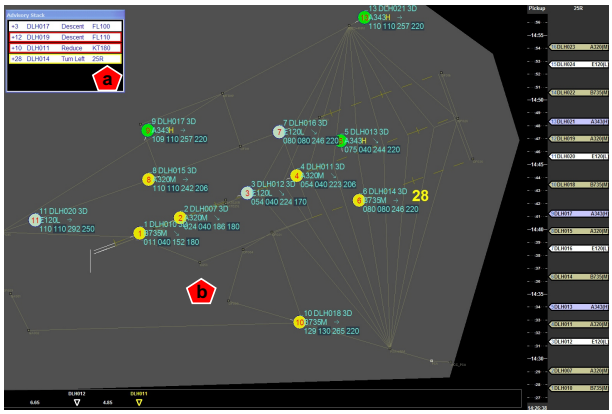

Figure 1: The AMAN software GUI [4] used by the participants in the experiment. Commands which are to be issued to aircraft appear on the advisory stack (labelled as *a*). This stack is positioned above an integrated flight information display (labelled as *b*), which displays radar and flight plan data as well as other information related to arrival management.

2.3. Grammar

A grammar corresponding to the Nuance grammar format [5] was written to match the ATC command format described in Figure 2. In this grammar, semantic concepts relevant to ATC (e.g. callsigns, command goals and values) are embedded in XML tags (e.g. *two two zero knots* ≅ `<airspeed> two two zero </airspeed> knots`). This allows these concepts to be easily parsed as a semantic template (see Figure 2 for an example) [6]. For recognition, the grammar is converted to a finite-state machine (FSM) in the AT&T format [7]. The XML tags are mapped to special phone symbols which do not correspond to real acoustic states (as it is typically done for homophones by adding phonetic disambiguation symbols `#1`, `#2`, etc.). All whitespace characters in the tags are converted to underscores ("_").

$$s\begin{bmatrix}\textsc{Callsign} & \texttt{ADR23} \\ \textsc{Command} & \mathit{turn}\begin{bmatrix}\textsc{Direction} & \texttt{left} \\ \textsc{Heading} & 250\end{bmatrix}\end{bmatrix}$$

Figure 2: A semantic template representing an ATC utterance (specifically, *Adria two three turn left heading two five zero*), where `ADR23` is the visual representation of the callsign.

3. Knowledge-Based Lattice Rescoring

When using a weighted finite state transducer (WFST) decoder [7, 8, 9], word lattices may be generated efficiently by 1.) creating a context-dependent phone-to-word transducer lattice, as shown in Figure 3, 2.) projecting onto the word labels and then 3.) performing epsilon removal and pruning [10]. Following Wölfel and McDonough [9], however, we perform the rescoring directly on the phone-to-word transducer lattice, due to the fact that the ASR toolkit we use does not currently allow the creation of weighted word lattices.

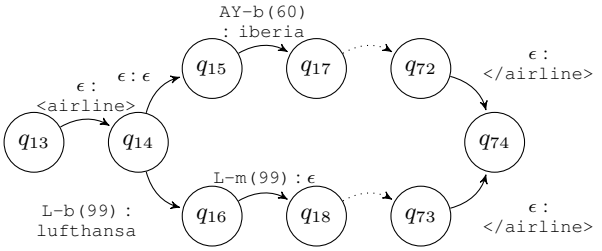

Figure 3: Part of a context-dependent phone-to-word transducer lattice [10] with embedded XML tags. The `L-b(99)`, `L-m(99)` and `AY-b(60)` symbols indicate triphone clustered acoustic states. The dotted edges indicate omitted nodes. Also note that there are costs associated to each of the edges although these are not shown here for the sake of readability.

3.1. Lattice Rescoring

Given a phone-to-word transducer lattice, the score of a particular speech recognition hypothesis $w = \langle \omega_1 \dots \omega_n \rangle$ is defined as the lowest-scoring path which outputs this sequence:

$$f(w) = \min_{\substack{\Sigma_1 \dots \Sigma_n \\ q_0 \dots q_{n-1}}} \sum_{i=1}^{n} \delta(q_{i-1}, \Sigma_i, \omega_i, q_i) \qquad (1)$$

where $\delta \colon Q \times (\Sigma \cup \{\epsilon\})^* \times (\Omega \cup \{\epsilon\}) \times Q \mapsto \mathbb{R}$ is the cost of reaching state q_i from state q_{i-1} under input of the phone input sequence $\Sigma_i = \langle \sigma_{(i,1)}, \dots, \sigma_{(i,n_i)} \rangle$ and output of the word ω_i plus an arbitrary number of ϵ symbols. In rescoring, the evaluation of the cost function δ is modified to incorporate additional knowledge sources after the actual recognition process (Viterbi search) by setting:

$$\delta(\cdot) = w_{\text{lm}}\delta_{\text{lm}}(\cdot) + w_{\text{am}}\delta_{\text{am}}(\cdot) + w_{\text{ks}}\delta_{\text{ks}}(\cdot) \qquad (2)$$

where δ_{lm} and δ_{am} denote the original language model (LM) and acoustic model (AM) scores (i.e. the log likelihood) from decoding and where δ_{ks} denotes the score under the additional knowledge source. The w_{lm}, w_{am} and w_{ks} denote the corresponding weights.

3.2. Knowledge Rescoring

Additional knowledge sources that have previously been used for rescoring include higher-order n-grams [11] and contextual articulatory knowledge [12, 13]. In this work, we consider rescoring as a means of penalizing hypotheses which are invalid or unlikely in the context in which the utterance was made: For example, in the ATC scenario from Section 2, the command *Lufthansa five one reduce speed two two zero knots* would be invalid if there is no aircraft in the airspace with that callsign.

Likewise, such a command would be unlikely if the aircraft did exist but is already flying at an airspeed of less than 220 knots.

In order to implement this notion of knowledge-based rescoring, we use semantic template structures [6] s which represent the meaning of the sentence or part of sentence in the lattice (see Figure 2). These structures are filled on the fly while traversing the lattice. Once a sub-structure z has been recognized completely (i.e. once the corresponding XML closing tag has been reached — see e.g. `</airline>` in Figure 3), a contextual *knowledge score* is applied. In this work, the knowledge score is calculated as the weighted sum of constraint penalty functions c_j:

$$\delta_{\text{ks}}(q_{i-1}, \Sigma_i, \omega_i, q_i) = \sum_{j=1}^{J} \rho_j c_j(x, z, s), \tag{3}$$

where $c_j(\cdot) \mapsto [0, 1]$ measures the degree to which a certain constraint c_j is violated given the semantic context of the sentence s processed so far as well as the discourse situation x (e.g. the current state of the airspace); The ρ_j denote penalty scores for violating the constraints. It is worth mentioning that this formulation is similar to the one given in Chang et al. [14], except that we use constraint programming for rescoring hypotheses while Chang et al. [14] use it for biasing inference during training.

3.3. Application to the ATC Task

For the ATC task, the context in which the dialog is situated can easily be obtained from AMAN software, such as 4D-CARMA [15] from the DLR. This software allows the retrieval of the current state of the airspace, including the callsigns of aircraft on the radar as well as their flight information such as airspeed, altitude and heading. As a proof of concept for knowledge-based rescoring, this information was used to apply the following (albeit relatively simple) semantic constraints:

- $c_{\text{callsign}}(\cdot)$, which is violated if the aircraft callsign denoted by a hypothesis does not refer to any aircraft in the airspace when the utterance was made.
- $c_{\text{spd}}(\cdot)$, which is violated if, given a command to reduce airspeed, the goal airspeed denoted by a hypothesis z_{spd} is not less than the last known airspeed of the aircraft denoted by the hypothesis h_{spd} ($c_{\text{spd}}(\cdot) = [\![z_{\text{spd}} \geq h_{\text{spd}}]\!]$).
- $c_{\text{alt}}(\cdot)$, which is violated if, given a command to descend to a lower altitude, the goal altitude denoted by a hypothesis z_{alt} is not less than the last known altitude of the given aircraft h_{alt} ($c_{\text{spd}}(\cdot) = [\![z_{\text{alt}} \geq h_{\text{alt}}]\!]$).

All these constraints are here used in their binary form, i.e. they return 1 if the constraint is violated and 0 otherwise.

4. Experiments

In this section, the usefulness of knowledge-based rescoring is evaluated through a set of automatic speech recognition (ASR) experiments. These experiments were performed on the ATC corpus described in the next section. The experimental setup is explained in Section 4.2, including a short description of the ASR system. Finally, the results are presented in Section 4.3.

4.1. The ATC Corpus Used

Due to the unavailability of an ATC corpus which includes the contextual information necessary for rescoring, a corpus dedicated to this purpose was recorded using the 4D-CARMA software [15] from the DLR. During the recordings, the participants observed a pre-scripted ATC simulation using an ATC arrival management (AMAN) software GUI client [4] (see Figure 1), which displayed simulation data for an airport landing scenario with 31 unique aircraft and one runway [15]. This included displaying ATC commands (in English), which the participants read aloud and which were recorded using a headset. The corresponding aircraft state vectors were retrieved every 5 seconds and then stored to a log file. Self-identified by first language, eight were German speakers, three were North American English speakers, and there were two Greek, one Malayalam, one Romanian and one Russian speaker. Twelve of the speakers were male and four were female. 1,107 ATC commands were recorded in total, with an average length of 9.5 words per sentence; This corresponds to approximately 100 minutes of speech. Each individual recorded utterance was annotated not only with the true sentence that was read but also with the state of the entire ATC simulation at the time of the recording (e.g. the callsigns of aircraft on the radar, including theirs speeds, altitudes, heading, position in relation to the radar, etc.).

4.2. Experimental Setup

The ASR system used for the experiments was the Millennium toolkit [9] in its standard configuration. Its feature extraction is based on Mel frequency cepstral coefficients that are obtained using warped MVDR spectral estimation [9]. After cepstral mean subtraction with variance normalization, 15 consecutive frames of 20-coefficient MFCCs are concatenated and subsequently reduced by linear discriminant analysis (LDA). The final features are 42-dimensional. The decoder is implemented along the lines of Saon et al. [8]. It generates phone-to-word transducer lattices as described in Ljolje et al. [10], which can then be used for rescoring. In this work, the rescoring was performed as described in Section 3, utilizing the radar data logged at the time each command was uttered. The best hypothesis was subsequently extracted from the rescored lattice.

The triphone acoustic model used in the experiments was trained on an amalgamation of several corpora of both spontaneous and scripted speech, including the Translanguage English Database and the WSJ0 and WSJ0CAM Wall Street Journal corpora as well as the ISL Meeting Speech and the NIST Automatic Meeting Recognition corpora. It has 4127 fully continuous codebooks with a total of 190063 Gaussians. The pronunciation lexicon used was the freely-available SPHINX dictionary plus some manually added pronunciation variants which allow for proper nouns (such as *Lufthansa* or *KLM*) to be recognized when they are pronounced in German. The recognition grammar is the one described in Section 2.3; It has a branching factor of 7.68, covers 244,220 unique aircraft callsigns and has a sentence perplexity of $3.9 \cdot 10^9$.

As evaluation metrics for speech recognition, we use both the word error rate (WER) and the sentence error rate (SER). In addition to these, we show the mean reciprocal rank (MRR), which is often used to measure the improvement through rescoring. It is calculated as:

$$\text{MRR}(Y) = \frac{1}{|Y|} \sum_{y \in Y} \frac{1}{\text{rank}(y)} \tag{4}$$

where Y denotes the complete set of utterances and where the reciprocal rank $\text{RR}(y) = 1/\text{rank}(y)$ of the gold standard transcription y is calculated by ranking each unique hypothesis in the word lattice by its score.

4.3. Results

Table 2 shows the speech recognition results before and after rescoring. The first row indicates the baseline WER, SER and MRR without rescoring ("none"). The second and third rows give the results after rescoring with the callsign constraint from Section 3.3 as well as with additional speed and altitude constraints for the individual aircraft. The last row ("oracle") shows the best possible results that could theoretically be obtained with an optimal rescoring algorithm.

Constraints Used	WER	SER	MRR
none (baseline)	2.81	22.58	0.849
callsign	0.55	4.61	0.966
callsign, spd, alt	0.52	4.52	0.967
oracle	0.31	2.07	0.979

Table 2: Word error rate, sentence error rate and mean reciprocal rank before and after rescoring with different constraints. The constraints penalize 1.) callsigns that are not on the radar as well as 2.) unlikely speeds and altitudes.

Most notable is that the callsign constraint gives a significant improvement in both word and sentence error rate, with relative reductions of 80.4% and 79.6% over the baseline. At the same time, the MRR is considerably higher, which means correct hypotheses get a better ranking in the n-best list. This can be explained by the fact that the callsign constraint effectively reduces the total number of possible callsigns from 244,220 to the number of aircraft which are currently in the airspace (in our scenario that is ≤ 10 at the time of the utterance). When additionally using dynamic speed and altitude constraints, the WER could be further reduced to 0.52, which corresponds to a total reduction of 81.49% over the baseline. The oracle results reveal that theoretical optimum rescoring can again cut the SER in half. Hence, there is still room for improvement.

5. Conclusions

In this paper, it has been demonstrated that the accuracy of automatic speech recognition can be greatly improved by rescoring ASR hypotheses based on contextual knowledge: In particular, this was shown in an ATC scenario in which the required contextual information could be extracted from radar data about the current state of the airspace. The declarative constraints used in these experiments were relatively simple, but they could easily be extended to incorporate the arrival manager's knowledge about likely approach paths in a certain ATC situation. As an alternative to rescoring, the knowledge constraints could also be integrated directly into the decoder, e.g. by creating the semantic templates (see Section 2.3) during decoding and then directly applying penalties based on the given constraints. This would allow for more aggressive pruning during beam search, which also could have potential performance benefits.

6. Acknowledgements

We would like to thank the Institute for Flight Guidance of the German Aerospace Center (DLR), especially Jürgen Rataj, Heiko Ehr, Oliver Ohneiser and Meilin Schaper for providing simulation software support as well as information about the ATC scenario and the domain of ATC in general.

7. References

[1] G.-J. M. Kruijff *et al.*, "Situated dialogue processing for human-robot interaction," in *Cognitive Systems*, ser. Cognitive Systems Monographs, H. I. Christensen *et al.*, Eds. Springer, Jul. 2010, vol. 8, pp. 311–364.

[2] 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*, Oct. 2004, pp. 169–172.

[3] *All Clear Phraseology Manual*, Eurocontrol, Brussels, Belgium, Apr. 2011. [Online]. Available: http://www.skybrary.aero/bookshelf/books/115.pdf

[4] H. Helmke, "Time-based arrival management," *Air Traffic Technology International*, pp. 40–43, 2011.

[5] *Nuance Speech Recognition System 8.5*, Nuance Communications, Inc., Dec. 2003.

[6] Y. Wang, L. Deng, and A. Acero, "Semantic frame based spoken language understanding," in *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, G. Tur and R. de Mori, Eds. Chichester, England, UK: John Wiley & Sons, May 2011, ch. 3.

[7] M. Mohri, F. Pereira, and M. Riley, "A rational design for a weighted finite-state transducer library," in *Automata Implementation*, ser. Lecture Notes in Computer Science, D. Wood and S. Yu, Eds. Springer, 1998, vol. 1436, pp. 144–158.

[8] G. Saon, D. Povey, and G. Zweig, "Anatomy of an extremely fast LVCSR decoder," in *Interspeech 2005*, Sep. 2005, pp. 549–552.

[9] M. Wölfel and J. M. McDonough, *Distant Speech Recognition*. Chichester, England, UK: John Wiley & Sons, Apr. 2009.

[10] A. Ljolje, F. Pereira, and M. Riley, "Efficient general lattice generation and rescoring," in *Eurospeech 1999*, Sep. 1999, pp. 1251–1254.

[11] T. Hain *et al.*, "Transcribing meetings with the AMIDA systems," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 486–498, Feb. 2012.

[12] J. Li, Y. Tsao, and C.-H. Lee, "A study on knowledge source integration for candidate rescoring in automatic speech recognition," in *ICASSP 2005*, vol. 1, Mar. 2005, pp. 837–840.

[13] S. M. Siniscalchi, J. Li, and C.-H. Lee, "A study on lattice rescoring with knowledge scores for automatic speech recognition," in *Interspeech 2006*, Sep. 2006, pp. 517–520.

[14] M.-W. Chang, L. Ratinov, and D. Roth, "Constraints as prior knowledge," in *ICML Workshop on Prior Knowledge for Text and Language Processing*, Jul. 2008, pp. 32–39.

[15] H. Helmke *et al.*, "Time-based arrival management for dual threshold operation and continous descent approaches," in *Proceedings of the Eighth USA/Europe Air Traffic Management Research and Development Seminar (ATM2009)*, 2009.