How to Measure Speech Recognition Performance in the Air Traffic Control Domain? The Word Error Rate is only half of the truth!

Hartmut Helmke[1], *Shruthi Shetty*[1], *Matthias Kleinert*[1], *Oliver Ohneiser*[1], *Heiko Ehr*[1], *Amrutha Prasad*[2], *Petr Motlicek*[2], *Aneta Cerna*[3], *Christian Windisch*[4]

[1] German Aerospace Center (DLR), Institute of Flight Guidance, Braunschweig, Germany
[2]Idiap Research Institute, Martigny, Switzerland,
[3]Air Navigation Service Provider, Jenec, Czech Republic, [4]Austro Control, Wien, Austria

hartmut.helmke@dlr.de, shruthi.shetty@dlr.de, matthias.kleinert@dlr.de, oliver.ohneiser@dlr.de, heiko.ehr@dlr.de, amrutha.prasad@idiap.ch, petr.motlicek@idiap.ch, cernaa@ans.cz, christian.windisch@austrocontrol.at

Abstract

Applying Automatic Speech Recognition (ASR) in the domain of analogue voice communication between air traffic controllers (ATCo) and pilots has more end user requirements than just transforming spoken words into text. It is useless, when word recognition is perfect, as long as the semantic interpretation is wrong. For an ATCo it is of no importance if the words of greeting are correctly recognized. A wrong recognition of a greeting should, however, not disturb the correct recognition of e.g. a "*descend*" command. Recently, 14 European partners from Air Traffic Management (ATM) domain have agreed on a common set of rules, i.e., an ontology on how to annotate the speech utterance of an ATCo. This paper first extends the ontology to pilot utterances and then compares different ASR implementations on semantic level by introducing command recognition, command recognition error, and command rejection rates. The implementation used in this paper achieves a command recognition rate better than 94% for Prague Approach, even when WER is above 2.5%.

Index Terms: word error rate, command recognition rate, language understanding, air traffic control, ATC

1. Introduction

Nowadays, enhanced Automatic Speech Recognition (ASR) systems are used in Air Traffic Control (ATC) training simulators to replace expensive pseudo pilots. This work has started already in the late 80s [1]. Although ASR systems are widely used in everyday life (e.g., Siri®, Alexa®) and ATC phraseology is standardized, recognizing and understanding controller-pilot communication is still a big challenge and not solved with satisfactory performance. Due to the lack of ATC-specific training data, current ASR systems still face challenges with specialized ATC vocabulary and syntax, controllers' deviations from the standard phraseology, and the variety of speakers, accents [2]. Cordero et al. (2012) reported WER (= word error rate) of more than 80% with different Commercial-off-the-shelf (COTS) recognizers [3].

Different metrics exist to evaluate the performance of ASR. The most widely used metric in ASR applications is the WER based on the Levenshtein distance [4]. However, the decision makers of air navigation service providers (ANSPs) are not primarily interested in these low-level metrics. They are interested in reducing costs and efforts. The AcListant®-Strips project quantified the benefits of using speech recognition with respect to both efficiency and ATCo workload: The workload for radar label maintenance by ATCo could be reduced by a factor of three [5] and the support of ASR enabled fuel savings of 50 to 65 liters per flight [6].

In this paper we will concentrate on the semantic level, i.e. on **annotations**, to evaluate the ASR performance in ATC domain, illustrated by the following two transcriptions for ATCo utterances:

- "good morning lufthansa two bravo alfa radar contact descend flight level eight zero and speed two two zero knots",
- "bravo alfa identified two twenty knots descend level eighty".

On word level there is a large difference between both transcriptions, but semantically they have a similar meaning. According to the ontology defined by various European partners from the ATM industry and research [7], both transcriptions correspond to the following three ATC commands: "DLH2BA INIT_RESPONSE, DLH2BA DESCEND 80 FL, DLH2BA SPEED 220 kt", but provided in a different order. The ontology rules enable the comparison of different speech recognition and understanding systems for ATC application: Consider each ATC command (e.g. DLH2BA SPEED 220 kt) as one (big) entity, i.e. word, and calculate the Levenshtein distances w.r.t. the gold annotations.

The following section introduces the main elements of the ontology for ATC command annotation. The ontology, introduced by the CWP HMI project [7] is not final yet, which means that updates/changes ae still expected. The projects STARFiSH [8] and "HMI Interaction Modes for Airport Tower" [1] e.g. expand the ontology with respect to ATC ground and tower commands including remote tower operations. Section 3 presents the suggested metrics for evaluation of speech recognition and understanding systems. Section 4 presents evaluation results from different projects, followed by the conclusions.

2. Ontology for annotation of ATC utterances

A subset of the CWP HMI ontology for annotation, which is being extended in the HAAWAII project is presented in this section. They define, that an utterance consists of one or more instructions (Figure 1) and each instruction starts with the callsign, even if the callsign is only said once. The full intended callsign (from the flight plan or surveillance data) is provided, i.e., AUA123B is used even if only "austrian three bravo" is said or recognized. This compensates for misrecognitions on word level and also deals with commonly used abbreviations for callsigns in ATC. If no callsign is said or could not be uniquely determined, "NO_CALLSIGN" is used. Figure 1 depicts the structure of an instruction and shows that an instruction consists of a callsign, a command and optional conditions. A command always has a type, which determines, how many values are allowed. Optional fields are unit (e.g., FL, ft, kt), qualifier (e.g., LESS, OR_BELOW, LEFT), speaker (PILOT or empty), and reason (REQUEST, REPORTING or empty for e.g. readbacks and commands).

Various examples from different application areas should illustrate the agreed rules. For approach traffic *"speed bird six nine six victor keep speed one six zero knots until four miles final"* would result in "BAW696V MAINTAIN SPEED 160 kt UNTIL 4 NM FINAL". The last four elements after "kt" are the conditional clearance with the conjunction "UNTIL". Here, we also see that the Type may consist of two words (e.g., MAINTAIN SPEED). An example from the tower is *"lufthansa four nine nine taxi to alfa five eight via lima and november eight"*, which results in "DLH499 TAXI TO STAND_A58" and "DLH499 TAXI VIA TX-L TX-N8". The command type "TAXI TO" can only have one value, whereas multiple values are allowed for "TAXI VIA". The ontology requires a configuration file, which defines that the word sequence *"alfa five eight"* is mapped to "STAND_A58" and that *"lima"* in a TAXI VIA command is mapped to "TX-L".

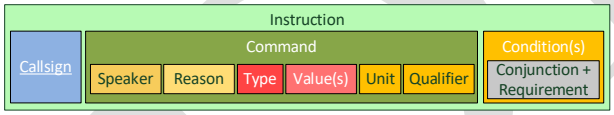

Figure 1: *Elements of an instruction consisting of a callsign, a command and condition(s).*

The following utterance considers both pilot and ATCo utterances for enroute traffic: *"Pilot: reykjavik control [NE Icelandic] godan dag [/NE] iceair six eight lima passing level one nine zero climbing two nine zero ATCO: [unk] six eight lima reykjavik control [NE Icelandic] godan dag [/NE] identified climb to flight level three seven zero"*. "NE" means "Non-English". The transcription rules require that the speaker names followed by a colon ("ATCO:" and "Pilot:") are added, if more than one speaker occurs in an utterance. The above utterance results in the annotation:

ICE68L PILOT STATION REYK_RADAR,
ICE68L PILOT GREETING,
ICE68L PILOT REPORTING ALTITUDE 190 FL,
ICE68L PILOT REPORTING CLIMB 290 none,
ICE68L STATION REYK_RADAR,
ICE68L GREETING,
ICE68L INIT_RESPONSE,
ICE68L CLIMB 370 FL.

We add the optional Speaker field in the annotations only, if the speaker is not the ATCo. If an altitude report or clearance does neither contain "feet" nor "flight level", the unit field is "none". Even if the full callsign uttered by the ATCo was not understandable or said, the annotation always contains the full callsign of an aircraft. The reason "REPORTING" is used for the pilot speaker only, if the altitude value is not a readback and is also not an altitude request. It is, however, not always decidable whether e.g., "… descending flight level two five zero" from the pilot is an altitude readback or a report. Both "ICE68L PILOT REPORTING DESCEND 250 FL" and "ICE68L PILOT DESCEND 250 FL" are, therefore, possible. One could easily determine which one is correct, by looking into the previous utterances. The annotation rules, however, require considering only the current utterance for creating the annotations. The transcription *"okay we check thanks air canada eight five four"* results in "ACA854 NO_CONCEPT" - not all words are covered by the ontology rules. *"okay we check thanks air canada eight five four descend three thousand feet"* would, however, result in "ACA854 DESCEND 3000 ft". NO_CONCEPT is extracted only if no other command type is extracted for this callsign. An implementation of the ontology from DLR already exists, which includes an automatic extraction (command recognition) from word sequences to the ontology concepts. In general, the command extraction first looks for fully matching callsigns followed by extraction of complete commands, incomplete commands (i.e., clearances given without known keywords), and values. The final step extracts non-fully matching callsigns from words not belonging to an already extracted command. More details are provided in [10]

3. Metric for semantic extraction accuracy

The user, e.g., the ATCo using speech recognition, is interested in a high recognition and a low error rate on sematic level, i.e., the meaning behind the spoken word sequence must be interpreted correctly. Quantifying the accuracy on semantic level, i.e., recognition accuracy and error rate, is described in this section. We use command recognition rate, command recognition error rate, and command recognition rejection rate, due to consistency with [11], Nevertheless, a wrong condition counts as an error in the command recognition error rate.

Command recognition rates are computed by comparing instructions from **manual** human annotation (gold annotation) to the results of the **automatic** semantic extraction (command extraction). For a given speech utterance, each instruction (see Figure 1) is treated as one big word. Then, the Levenshtein distance between the gold annotation and the results of command extraction is calculated, resulting in the number of substitutions (subs), insertions (ins) and deletions (del). Table 1 gives an overview about the different metrics and illustrates an example how they are calculated. In the table #gold defines the total number of commands in the gold annotation. #match defines the number of matches, which is #gold – subs – del. If the result of the command extraction contains either NO_CONCEPT or NO_CALLSIGN, these substitutions and insertions are always calculated as deletions, i.e., these extractions contribute to the rejection rate and not to the error rate (as shown in the example in Table 1).

For calculation of the callsign rates CaR, CaE and CaRj we just compare the callsigns from the gold annotation and from the automatic extraction. For each utterance we consider the callsign only once, except when different callsigns are

annotated or extracted. For the example in Table 1 this results in the three annotated and extracted callsigns AFR123, AUA1AB and DLH123.

Table 1: *Metric definition.*

Metric	Calculation
Command Recognition Rate (**RcR**)	**RcR** = #matches / #gold
Command Recognition Error Rate (**ErR**)	**ErR** = (subs + ins) / #gold
Command Rejection Rate (**RjR**)	**RjR** = del / #gold
Callsign Recognition Rate (**CaR**)	Same as RcR but only for callsigns without instructions
Callsign Recognition Error Rate (**CaE**)	Same as ErR, but only for callsigns without instructions
Callsign Rejection Rate (**CaRj**)	Same as RjR, but only for callsigns without instructions
If the command extraction results in different callsigns, the calculation is done for each callsign. See example below, which also illustrate that the sum of RcR, ErR and RjR can exceed 100%.	
Example	
Gold Annotation	**Command Extraction**
AFR123 INIT_RESPONSE AFR123 TURN LEFT AUA1AB SPEED 140 kt DLH123_NO_CONCEPT	AFR123 DIRECT_TO OKG none AFR123 INIT_RESPONSE AFR123 TURN RIGHT AUA1AB NO_CONCEPT DLH123 NO_CONCEPT
Result:	
RcR = 2/4 = 50% (green)	*ErR = 2 / 4 = 50% (purple)*	*RjR =1/4 = 25% (yellow)*

As the ontology is still evolving, the annotations and extractions for different data sets are based on different versions of the ontology. In most cases new ontology versions introduce new command types. The metric calculation has to take this into accounts so that older data sets can also be reused. If some command types were not considered in the gold annotation or by the extraction (set via a configuration file), these command types are deleted from both the annotation and from the extraction. If after the deletions the set of annotations or extraction for a callsign is empty, the command type NO_CONCEPT is added for this callsign. If INIT_RESPONSE and SPEED command types are not supported for the above example from the metric definition this would lead to the following result as shown in Table 2.

Table 2: *Example of metric definition with INIT_-RESPONSE and SPEED commands switched off.*

Gold Annotation	Command Extraction
AFR123 TURN LEFT AUA1AB NO_CONCEPT DLH123 NO_CONCEPT	AFR123 DIRECT_TO OKG none AFR123 TURN RIGHT AUA1AB NO_CONCEPT DLH123 NO_CONCEPT
AFR123 INIT_RESPONSE is mapped to AFR123 NO_CONCEPT. However, both gold annotation and command extraction contain still another command for AFR123. NO_CONCEPT is only added, if it is the only command, which is here the case for AUA1AB with SPEED mapped to NO_CONCEPT.	
Result:	
RcR = 2/3 = 67% (green)	*ErR = 2/3 = 67% (purple)*	*RjR = 0 = 0% (yellow)*

4. Experimental results

Trials with data from ANSPs of Prague and Vienna from the *MALORCA* and *CWP HMI* project were performed. For both Prague and Vienna gold transcriptions and gold annotations of the ATCo voice recordings were available. From simulation runs (Lab) of the *CWP HMI* project 6,885 commands were taken from five different ATCos from Prague and 6,005 commands were taken from two different ATCos from Vienna ATCos (see rows with *Labs*) [12]. From the MALORCA project 6,094 from Prague approach and 4,417 commands from Vienna approach were taken from operational environment recordings of 12 and 41 ATCos [13], respectively (see rows with *Ops*). The number of commands per speech utterance was between one and seven.

Table 3: *Recognition accuracy for ops room and lab.*

	#Cmd	#Utt	RcR	ErR	CaR
Ops Prague	6094	3038	98.5%	0.9%	99.8%
Lab Prague	6885	4211	99.2%	0.5%	99.7%
Ops Vienna	4417	2279	94.8%	4.0%	98.2%
Lab Vienna	6005	3562	95.3%	2.5%	96.4%

Table 3 shows the metrics, number of commands (#Cmd) and speech utterances (#Utt) for the different data sets. The command extractions in this table are performed on the gold transcriptions (WER=0%) and, therefore, shows the upper limit of command extraction if the word recognition is perfect. More interesting are the results, when the output from a speech-to-text engine with WERs > 0% is used. For the results of Table 4 different models and context information were used, which led to different WER rates.

Table 4: *Recognition accuracy with different WERs.*

	RcR	CaR	WER
Ops Prague, gold transcription	98.5%	99.8%	0.0%
Ops Prague, no callsign context	96.5%	98.7%	2.3%
Ops Prague, callsign context	96.6%	98.2%	2.8%
Ops Prague, bad speech model	76.8%	88.5%	13.5%
Ops Vienna, gold transcription	94.8%	98.2%	0.0%
Ops Vienna, no callsign context	89.9%	93.0%	5.1%
Ops Vienna, callsign context	88.6%	91.6%	6.7%
Ops Vienna, bad speech model	82.7%	87.8%	9.5%

From Table 4, we see that a lower WER of 2.3% results in a worse command recognition rate (96.5%) as compared to a WER of 2.8%. The latter WER is based on using the context information, i.e., information regarding, which aircraft callsigns are currently controlled by the ATCo. The gold transcription “austrian two three one” is then recognized as e.g., “austrian three three one” if only AUA331 is available in context, although the ATCo clearly said “two three”. The rows with “bad speech model” in Table 4 refer to an ASR engine adapted by transcribed utterances from just one speaker, but which was used to recognize speech also from other speakers. This, of course, results in a bad performance concerning WER. The data does not only show that a lower WER not automatically results in a higher command recognition rate, but it also shows that fully recognizing an instruction/command does not require each word of the command to be correctly recognized. The command extraction algorithm always uses the information, which callsigns are currently in the air,

independent of the fact, whether the speech-to-text block is using this information or not.

Table 5: *Command length*.

WER	3	4	5	6	7	8	9
2.3%	93%	91%	89%	87%	85%	83%	81%
2.8%	92%	89%	87%	84%	82%	80%	78%
5.1%	85%	81%	77%	73%	69%	66%	62%
6.7%	81%	76%	71%	66%	62%	57%	54%
9.5%	74%	67%	61%	55%	50%	45%	41%
13.6%	65%	56%	48%	42%	36%	31%	27%

Table 5 shows what command recognition rates could be expected for certain WER and different average command length in words, if the WER would directly translate to the command recognition rate. Assuming that the sequence of words "descend flight level two one zero" consisting of six words only results in "DESCEND 120 FL" if all six words are correctly recognized, should result in a command extraction rate of 55% given a WER of 9.1%. The average command length for Prague data was 7.0 words and for Vienna 5.6 words. So, for a WER of 2.8% a command recognition rate of at most 82% should result, but we have achieved 96.6% (as shown in Table 4). Similarly, for a WER of 5.1% for Vienna ops room data without using callsign information from the surveillance data, we expect a command recognition rate of only 75%, but we observed 89.9%. The command extraction algorithm is quite robust.

Table 6 illustrates the results if we concentrate on altitude changing command types (column *DESCEND*) and direction changing command types (column *Direct to*), which are important in the ATC world. The upper part shows the results for Prague and the lower part for Vienna ops room data. The extraction rate for the DESCEND command decreases only slightly with increasing WER for *acceptable* WER, i.e. less than 7%, but heavily increases also for DESCEND command if WER gets worse. In those case we get also for the DESCEND command not better results than the average command recognition rates averaged over all command types.

Table 6: *Specific command recognition rates.*

Ops Prague			
WER	**All**	**DESCEND**	**DIRECT_TO**
	6063	925	370
0.0%	98.5%	99.8%	97.0%
2.3%	96.5%	98.3%	95.1%
2.8%	96.6%	99.0%	87.8%
13.6%	76.8%	76.1%	77.3%
Ops Vienna			
WER	**All**	**DESCEND**	**DIRECT_TO**
	4417	679	387
0.0%	94.8%	98.5%	91.0%
5.1%	89.9%	95.9%	86.6%
6.7%	88.6%	95.4%	82.2%
9.5%	82.7%	86.5%	77.3%

5. Conclusions

The paper has extended the ontology developed by SESAR solution CWP HMI also for pilot utterances. The implementation of the ontology rules results in command recognition rates of 99% for Prague airport and achieves 95% for Vienna airport. For Vienna the gold annotations are still improvable and the used phraseology contains a high variability often deviating from published standard phraseology. The implementation is robust against errors resulting from speech-to-text transformation. WER below 3% decreases performance only slightly. WER above 10% still enable command recognition rates better than 75%, even though the average command length was longer than 6 words. The command recognition rate metric is of course not new. The transformation of an ATC utterance consisting of a sequence of words into its semantic elements, based on the ontology, however, is new. Only the presented definition and the implementation of the extended ontology, enable a detailed comparison of different speech recognition and understanding applications also on semantic level and not just on word level. Using only the word error rate would be only half of the truth, but also not less. WER analysis provides initial hints with respect to the ASR performance.

6. Acknowledgements

The ATM community like to thank the organizers of the INTERSPEECH2021 for the opportunity to have a special session on "Automatic Speech Recognition in Air Traffic Management (ASR-ATM)". Three SESAR2020 industrial research projects PJ.16-04-W1 (CWP HMI), PJ.10-96-W2 (HMI Interaction modes for Approach control), and PJ.05-97-W2 (HMI Interaction Modes for Airport Tower) and the exploratory research project HAAWAII have received funding from the SESAR Joint Undertaking under the European Union's grant agreement No. 734141, 874464, 874470, and 884287.

7. References

[1] C. Hamel, D. Kotick, and M. Layton, "Microcomputer System Integration for Air Control Training.", Special Report SR89-01, Naval Training Systems Center, Orlando, FL, USA, 1989.

[2] Said, M. Guillemette, J. Gillespie, C. Couchman, and R. Stilwell, "Pilots & Air Traffic Control Phraseology Study," in International Air Transport Association, 2011.

[3] 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

[4] V.I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," Soviet Physics—Doklady 10.8, Feb. 1966.

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

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

[7] H. Helmke, M. Slotty, M. Poiger, D.F. Herrer, O. Ohneiser et al., "Ontology for transcription of ATC speech commands of SESAR 2020 solution PJ.16-04," IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), London, United Kingdom, 2018.

[8] STARFiSH, research project funded by the German Federal Ministry of Education and Research, see for further information https://www.softwaresysteme.pt-dlr.de/de/ki-in-der-praxis.php, in German, n.d.

[9] PJ.05-97-W2 SESAR2020 funded industrial research projects under the European Union's grant agreement 874464, see for further information https://www.remote-tower.eu/wp/?page_id=888, and https://www.remote-tower.eu/wp/?p=824 and https://www.sesarju.eu/index.php/projects/DTT, n.d.

[10] H. Helmke, M. Kleinert, O. Ohneiser, H. Ehr, S. Shetty, "Machine Learning of Air Traffic Controller Command Extraction Models for Speech Recognition Applications," IEEE/AIAA 39th Digital Avionics Systems Conference (DASC), San Antonio, Texas, USA, 2020.

[11] H. Helmke, J. Rataj, T. Mühlhausen, O. Ohneiser, M. Kleinert, Y. Oualil, and M. Schulder, "Assistant-based speech recognition for ATM applications," 11th USA/Europe Air Traffic Management Research and Development Seminar (ATM2015), Lisbon, Portugal, 2015.

[12] M. Kleinert, H. Helmke, H. Ehr, C. Kern, D. Klakow, P. Motlicek, M. Singh, and G. Siol, "Building Blocks of Assistant Based Speech Recognition for Air Traffic Management Applications," 8th SESAR Innovation Days, Salzburg, Austria, 2018.

[13] M. Kleinert, H. Helmke, S. Moos, P Hlousek, C. Windisch, O. Ohneiser, H. Ehr, and A. Labreuil, "Reducing Controller Workload by Automatic Speech Recognition Assisted Radar Label Maintenance," 9th SESAR Innovation Days, Athens, Greece, 2019.