Radboud University

Bachelor Thesis

Automatic Speech Recognition and Call Sign Detection on Air Traffic Control Data

Author:
Elena Kreis[1]
s4841670

Supervisor:
Dr. Franc A. Grootjen[1]

A thesis presented for the degree of
Bachelor of Science in Artificial Intelligence

Reading Committee:
Dr. Franc A. Grootjen[1]
Dr. Umut Güçlü[1,2,3]

[1]Department of Artificial Intelligence
[2]Donders Institute for Brain, Cognition and Behaviour
[3]Donders Centre for Cognition
Radboud University

July 20, 2020

Acknowledgments

I would like to thank Franc Grootjen for his insight and guidance with this thesis. I would also like to thank my family and friends for their loving support.

Abstract

Air traffic control (ATC) is an area of work with high workload for pilots and controllers. Automatic speech recognition (ASR) and call sign detection (CSD) could help reduce this workload. There are many pre-existing systems of automatic speech recognition, such as the sequence-to-sequence time-depth separable architecture by Hannun et al. (2019). Not many studies exist on call sign detection, one of the few being the Airbus challenge, where the team of Gupta et al. (2019) applied a bi-directional long short term memory and conditional random field classifier architecture. This study aims to investigate how these pre-existing systems perform on a different corpus of ATC data, and further, how errors of either system affect the final accuracy of the systems in sequence. The corpus investigated here is the *Air Traffic Control Communication* (ATCC) corpus by Šmídl (2011).

The ASR and CSD systems were trained on the ATCC data and tested both individually and in combination. The research showed that ASR performs poorly with a word error rate of 33.08%, while CSD achieves a good F-score of 0.8509, and the combined systems' performance is again quite poor with an F-score of 0.4931. The results indicate that the combined systems' performance is strongly influenced by transcription errors of the ASR system, more so than by errors of the CSD system.

Contents

List of Figures

List of Tables

List of Abbreviations

ASR	Automatic speech recognition
ATC	Air traffic control
ATCC	Air traffic control communication
Bi-LSTM	Bi-directional long short term memory
CoNLL	Conference on Computational Natural Language Learning
CRF	Conditional random field
CSD	Call sign detection
EOS	End of sentence
ICAO	International Civil Aviation Organization
LM	Language model
NATO	North Atlantic Treaty Organization
NER	Named entity recognition
SGD	Stochastic gradient descent
TD	Telephony designator
TDS	Time-depth separable
WER	Word error rate

1 Introduction

Technologies involving automatic speech recognition (ASR) have the potential to be of great help in air traffic control (ATC). Air traffic control is an area with high cognitive workload and high possible damage in case of a mistake. Pilots have to constantly listen to a stream of information on the radio, waiting for instructions directed at their aircraft. They furthermore have to memorize directions that can be both long and complicated. Controllers have to keep track of the directions they give to aircraft either on paper or via mouse input, which comes in addition to their already demanding task of coordinating and navigating several aircraft. Automatic speech recognition can help reduce workload for both pilots and ground control, by aiding or even replacing some of the tasks mentioned above. Recent studies (Xiong et al., 2017; Saon et al., 2017) have shown that automatic speech recognition can achieve human parity on the Switchboard corpus, which consists of conversational telephone speech. Another corpus, widely used in automatic speech recognition studies, is the Librispeech corpus. One model that has been applied to this corpus, is by Hannun et al. (2019), which is a sequence-to-sequence architecture using time-depth-separable convolutions. This model achieved word error rates (WER) of 5.36% (without a language model) and 3.28% (with a convolutional language model) on the 'clean' test set of Librispeech and 15.64% (without a language model) and 9.84% (with a convolutional language model) on the 'other' test set. Unfortunately, the characteristics of air traffic control data are quite different from corpora such as the Switchboard or Librispeech corpus. ATC data tends to be rather noisy and of bad quality, as it comes from VHF (very high frequency) radio communication, and speakers are often non-native, making it more challenging to achieve high accuracy speech recognition on ATC data. Therefore it appears interesting to investigate how the ASR system of Hannun et al. (2019) performs on ATC data.

1.1 Call Signs and Call Sign Detection

A crucial part of air traffic control are call signs, which are the unique identifiers used to refer to individual flights. They are required on the pilot's side in order to identify their aircraft when making radio contact with the tower, and controllers use them to direct instructions at aircraft over radio. Call signs have a specific format defined by the International Civil Aviation Organization and consist of the telephony designator of the airline, followed by numbers and letters spoken according to the NATO phonetic alphabet. An example of such a call sign is 'LUFTHANSA TREE NINER HOTEL' (Lufthansa 3 9 H). Correctly understanding call signs in air traffic control communication is crucial to avoiding confusion of instructions and therefrom resulting accidents. Especially in situations where flights with similar call signs are present in the same airspace or even on the same radiotelephone frequency, there is a high risk of call sign confusion and resulting misinterpretation of instructions. Several incidents have occurred, where call sign confusion was a causing factor, often due to an aircraft mistakenly accepting an instruction meant for another aircraft, such as take-off or flight level clearance.[1] To help prevent or mitigate such incidents we can apply call sign detection (CSD), which is the process of extracting call signs from text data, i.e. recognizing the call sign in a given sentence. In combination with automatic speech recognition, such a call sign detection system is applied to the transcript output by an ASR system. This way it can become a further addition to a system that helps reduce workload for pilots and controllers and can be used to help prevent accidents due to call sign confusion, for example by giving a warning when an instruction is accepted and read back by the wrong flight.

Call sign detection can be handled as a form of named entity recognition (NER). This is a type of information retrieval which aims to detect and classify named entities, such as person names, organizations, locations, etc., in a body of text. State-of-the-art systems can achieve named entity recognition at fairly high accuracies of over 90% on plain text (Lample et al., 2016). Unlike regular named entity recognition, which is mostly applied to human-written text, call sign detection is more naturally applied to transcriptions generated by an ASR system. Any errors in the transcription therefore can propagate to errors in the call sign detection, making it more challenging than regular NER. Furthermore, research on call sign detection is still fairly limited.

1.2 Airbus Challenge

As part of a challenge organized by Airbus in 2018 (Pellegrini et al., 2019), several teams competed to perform automatic speech recognition and call sign detection on ATC data. Overall the challenge produced remarkable results, with the best teams achieving word error rates between 8% and 10% for the ASR task and F-scores above 80% for the CSD task. The runner-up solution for the CSD task was the system by *Centre de Recherche Informatique de Montréal* (CRIM). Their call sign detection system used bi-directional LSTMs followed by a conditional random field classifier, which achieved an F-score of 0.8017 on the evaluation set (Gupta et al., 2019). To the knowledge of the author, the Airbus challenge is the only collection of research on call sign detection, making for no comparable results on other ATC data. There tends to be a large variability between corpora of ATC data, due to, for example, differing accents or audio quality. Hence it seems interesting to research whether a combined system of automatic speech recognition and call sign detection, using the systems of Hannun et al. (2019) and Gupta et al. (2019), can work well on other ATC corpora, possibly of lower quality.

Both tasks of the Airbus challenge were performed on a corpus of transcribed ATC data collected by Airbus (Delpech et al., 2018). A subset of this corpus was made available to participants in light of the challenge, but the data is not publicly available. To the knowledge of the author, there are no publicly available ATC corpora that include call sign annotations, a characteristic necessary

[1] `https://www.skybrary.aero/index.php/Call-sign_Confusion#Accidents_and_Incidents`

for performing call sign detection. One corpus mentioned in the Airbus challenge is the *Air Traffic Control Communication* (ATCC) corpus (Šmídl, 2011) which consists of Czech ATC data. It is publicly available and contains real-life non-native ATC speech, making it a good candidate data set. However, it has a rather small size of 20h and contains no call sign annotations. Although the small size is not beneficial for model training, it does make it feasible to manually annotate call signs in the frame of this research.

1.3 Research Question

This research will aim to investigate whether similar accuracy as that seen in the Airbus challenge can be achieved on a different data set, with different characteristics. More explicitly, the research question is:

> How will the ASR system of Hannun et al. (2019) and the CSD system of Gupta et al. (2019) perform on the ATCC corpus, in terms of WER and F-score respectively, both separately and in sequence?

and further,

> How does the interaction between the ASR and CSD systems affect the performance accuracy, i.e. how do errors of either system affect the final accuracy when the systems are applied to input in sequence?

A first step will be to annotate call signs in the aforementioned *Air Traffic Control Communication* corpus (Šmídl, 2011). To this data we then apply the sequence-to-sequence TDS architecture for speech recognition of Hannun et al. (2019), followed by the Bi-LSTM and CRF system for call sign detection by Gupta et al. (2019).

2 Methods

2.1 *Air Traffic Control Communication* Speech Corpus

The data set used in the Airbus challenge was made available to the teams solely for the purpose of the challenge and hence, is not publicly available. To investigate whether pre-existing systems for automatic speech recognition and call sign detection would work on other data as well, another corpus was chosen for this research. Here, the data used for training and evaluating in both ASR and CSD was taken from the *Air Traffic Control Communication* speech corpus (Šmídl, 2011), which consists of 20h of air traffic control speech, gathered in the Czech Republic. This corpus was chosen since it is publicly available and it contains real-life non-native English. The corpus consists of 2657 transcription files, along with their respective audio files, which contain 14832 speech turns. There are some transcription errors and inconsistencies in the corpus, such as spelling mistakes, and inconsistent notation of call signs (e.g. ‘Beeline’, ‘Bee Line’, ‘Bline’). Rozenbroek (2020) processed around 70%

of the data (1817 files) and corrected any mistakes or inconsistencies, and this improved data is used here alongside the other, unimproved, 30% of the data. A script is used to fix as many remaining mistakes and inconsistencies as possible. Furthermore, any audio files longer than five seconds are sliced up into speech turns prior to being used in the ASR system. This corpus does not contain call sign annotations, but more importantly, there are, to the author's knowledge, no publicly available ATC corpora that contain annotated call signs. A portion of the data was processed manually to add call sign annotations, to make it usable for this research. Specifics on that follow in the next section.

2.2 Call Sign Annotation

As explained in the previous section, it was necessary to annotate call signs in the ATCC corpus, so as to have data to train and evaluate the call sign detection system. Due to limited time, only a third of the files were annotated, totaling at 937 files. This portion of data contains 4438 call signs, of which 1085 are unique. These numbers are excluding incomplete or incorrect call signs. There are an additional 253 incomplete call signs, 190 of which are unique. More information on what classifies as an incomplete or incorrect call sign follows below. The corpus contains mostly call signs of type C and a few of type A and B, according to the ICAO definition of call signs (ICAO, 2001). Only call signs of type C were annotated in the data since they are the most common and were hence chosen to be the focus of this research. Examples of each type are depicted in Table 1.

Type of CS	Example
Type A	VPBDX
Type B	RSY9715
Type C	LUFTHANSA 787

Table 1: Types of call signs and examples of them from the ATCC corpus

The ATCC corpus uses square brackets for annotations, with ranges being noted in the following format:

```
[annotationtype_|] transcription [|_annotationtype]
```

Therefore the annotation chosen for call signs was:

```
[cs_|] call sign [|_cs]
```

There are some incomplete or otherwise incorrect call signs in the data. Often the telephony designator of the airline is left out, for example, '7 8 7' is said instead of, the correct, 'LUFTHANSA 7 8 7'. These call signs, along with instances of just telephony designators without any identification number, were considered incomplete. Call signs were deemed incorrect if, for example, they were said incorrectly and immediately corrected, resulting in a format not conforming to the ICAO definition. For example ‘AIR BERLIN 4 6 1 3’ is incorrectly said as ‘AIR BERLIN 3 [ehm_??] 4 6 1 3’. On the other hand, call signs that were incorrect in context, say a digit replaced by another digit, but correct in format, were not tagged as incorrect, but as a regular call sign since the ‘incorrect’ tag is intended to refer to incorrect format. For example instead of ‘LUFTHANSA 7 8 7’, the controller said ‘LUFTHANSA 7 6 7’. Any incomplete or incorrect call signs were given the tag:

```
[ics_|] incomplete/incorrect call sign [|_ics]
```

The ASR system was trained to identify both regular and incomplete or incorrect call signs as call signs. The choice not to distinguish between the two was made since incomplete call signs occur quite regularly in ATC communication.

The adapted corpus containing call sign annotations can be found on GitLab[2].

2.3 ASR System

The ASR system was based on the speech recognition system by Hannun et al. (2019) which consists of a sequence-to-sequence encoder with time-depth separable (TDS) convolutions. This system is a fully convolutional architecture, based on TDS blocks that consist of 2D convolutions over time, with a subsequent fully connected layer, as illustrated in Figure 1.

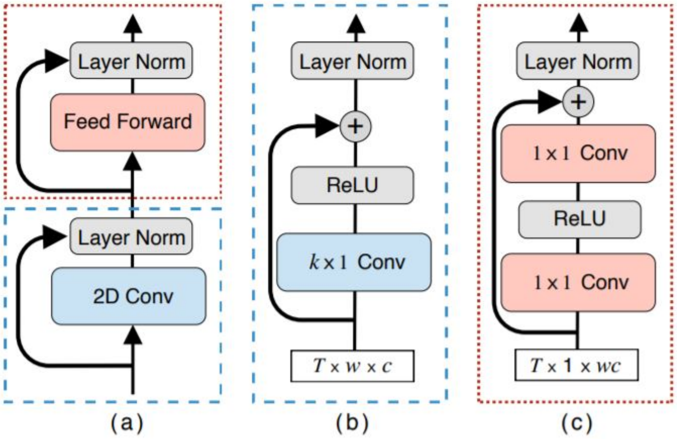


Figure 1: ‘The TDS convolution model architecture. (a) The sub-blocks of the TDS convolution layer are (b) a 2D convolution over time followed by (c) a fully connected block’, where ‘T is the number of time-steps, w is the input width and c is the number of input (and output) channels.’ (Hannun et al., 2019)

.

[2] https://gitlab.socsci.ru.nl/atc-and-nlp/CzechATCC. The master branch contains data corrected by Rozenbroek (2020) and the callsign branch contains data with call sign annotations. This repository is currently not publicly available.

The network architecture was slightly modified from the original, by increasing the number of channels in the TDS blocks, but the general structure remained the same. Credit for tuning this architecture as well as hyperparameters goes to Rozenbroek (2020). The final encoder architecture consisted of two 12-channel, three 16-channel, and six 20-channel TDS blocks, with three 1-D convolutional layers before each group of TDS blocks, each with kernel size 25 x 1. At the end of the encoder, a linear layer was used to generate the 1024-dimensional output. For the full architecture file with all parameters, see Appendix A.1.

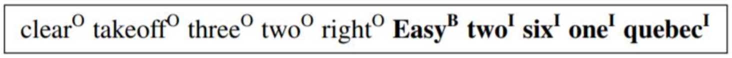
For acoustic front-end, 80-dimensional mel-scale filter banks were used, computed with a bin size of 25 ms. The optimizer used was SGD, with a learning rate of 0.05 and a decaying factor of 0.5. Gradients were clipped to 15 and the model was pre-trained with the soft window and $\sigma = 4$. Dropout was at 0.2, label smoothing at 0.05, and random sampling at 0.01. For the full list of flags used for training, see Appendix A.2.

The language model that was used for beam search decoding was a 3-gram language model, trained by Rozenbroek (2020). The decoding used a beam size of 80, an EOS penalty of 1.5, and $\eta = 10$. The LM weight was 1.0 and the word insertion score was 2.0. The EOS score was the only parameter changed from the original decoder setup, with a value of -3. Appendix A.3 contains the full list of flags used for decoding. The system was evaluated both with and without the beam search decoder, the latter by simply computing the greedy path through the acoustic model.

The wav2letter++ framework was used to train and evaluate the model (Pratap et al., 2019). All 20h of speech from the ATCC corpus were used to train and evaluate this system.

2.4 CSD System

For call sign detection the system used was inspired by CRIM's solution to the Airbus challenge. This system treats call sign detection as a problem of named entity recognition and consists of a bidirectional LSTM (Bi-LSTM), paired with a conditional random field classifier (CRF). The data was pre-processed to have a set of associated tags from the Inside-Outside-Beginning (IOB) tag set, where each word in the data gets label 'B' (beginning of call sign), 'I' (inside call sign), or 'O' (outside call sign), as shown in Figure 2.



clear^{O} takeoff^{O} three^{O} two^{O} right^{O} **Easy**^{B} **two**^{I} **six**^{I} **one**^{I} **quebec**^{I}

Figure 2: Example of data tagged using IOB tag set (Gupta et al., 2019)

The Bi-LSTM acts as a feature extractor, which gets word embeddings as input and returns the emission scores. These features are then passed onto the CRF, which uses a matrix of transition probabilities and the Viterbi algorithm to decode the probabilities returned by the feature extractor, and compute the

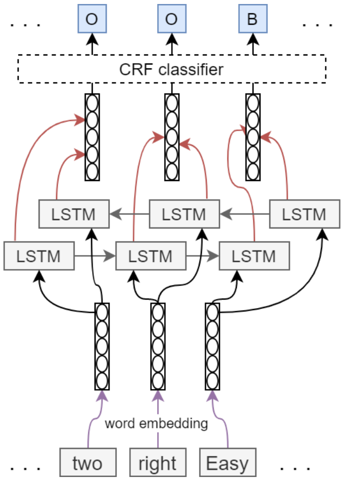


Figure 3: The Bi-LSTM - CRF architecture

most likely sequence of tags associated with the data. Figure 3 illustrates this architecture. The loss function used was negative log-likelihood and Adam was used as an optimizer, with a learning rate of 0.01 and a weight decay of 1e-4. A validation set was used to check the system's performance during training and the model with the highest validation accuracy was used for testing. The implementation of this system was done using PyTorch and largely based on a tutorial using the same architecture.[3]

To evaluate the system's performance, the F-score measure was used. Specifically, it was applied to entity level, not to token level, and the CoNLL standard on NER was used to decide when a call sign is deemed to be correctly tagged (Sang and Meulder, 2003). With this standard, a call sign is only counted as correct if it is an exact match to the true call sign. The CSD system was trained and tested on only the portion of the ATCC corpus that was processed with call sign annotations. Furthermore, it was also tested on this same data but with call signs replaced by randomized call signs, to assess generalizability to unseen call signs. Table 2 shows the two ways in which the randomizations were done, first, by keeping the telephony designators (TD) and simply randomizing the letters and digits, which ensures mostly that the test set does not contain any call signs already seen by the model in the train set while keeping the telephony designators familiar; and secondly, by randomizing letters and digits, as well as selecting a new telephony designator, not seen before in the training set. For the new telephony designators, a set of 40 telephony designators (see Appendix B), not occurring in the ATCC corpus, was chosen from ICAO (2017). For simplicity, only call signs with a telephony designator consisting of a single word, were used for this.

[3] `https://pytorch.org/tutorials/beginner/nlp/advanced_tutorial.html`

Original call sign	ADRIA 9 3 W
Randomized call sign, same TD	ADRIA 2 7 B
Randomized call sign, new TD	GLOBETROTTER 5 2 T

Table 2: Example of randomizing call signs to generate 'new' data.

2.5 Combining ASR and CSD

The two systems were trained and tuned separately. Evaluation of the systems was done in two different ways; each system separately, and the two systems in sequence. To evaluate the combination of the two systems, the transcription generated by the ASR system was input into the CSD system, which then outputs a set of call signs that were compared to the ground truth call signs in that sentence. Therefore the evaluation of the combination of the two systems was done using the F-score. To further assess the interaction between the two systems, it was calculated how many call signs were correctly transcribed by the ASR system, to give an indication of how much that could have limited the CSD's performance since the percentage of call signs correctly transcribed by the ASR is a direct limit on the recall that the CSD system can achieve when applied to that transcription.

The code used for call sign detection, as well as some scripts used for both ASR and CSD can be found on GitLab.[4]

3 Results

3.1 ASR

The lowest validation WER during training of the encoder was 31.13%. As mentioned in section 2.3, the ASR system was evaluated both with beam search decoding and without it. When using the decoder, the system performed very poorly, with the lowest WER achieved on the test set being 95.18%. When using just the greedy path through the acoustic model, the test WER was much closer to the validation WER, with 33.08%. Therefore beam search decoding was not used further in this study.

3.2 CSD

Using the model with the highest validation F-score (0.89) during training, the call sign detection system performs on the test set with an F-score of 0.8509. Notably, precision, with 0.9073 is quite a bit higher than recall, with 0.8011.

Using randomly generated call signs, the system scores 0.8133 for call signs with known telephony designators, with precision 0.8654 and recall 0.7670, again precision being higher than recall. For call signs with unknown telephony des-

[4] https://gitlab.socsci.ru.nl/E.Kreis/thesis-asr-and-csd

ignators, the F-score is 0.6951, where precision is 0.9299 and recall is 0.5550, here too precision is higher than recall. The values are summarized in Table 3.

	F-score	Precision	Recall
Test set, original	0.8509	0.9073	0.8011
Test set, random CS, known TD	0.8133	0.8654	0.7670
Test set, random CS, unknown TD	0.6951	0.9299	0.5550

Table 3: Performance of CSD system. CS stands for call sign. TD stands for telephony designator

3.3 Combination ASR & CSD

Using the transcription generated by the ASR system without beam search decoding, the call sign detection achieves an F-score of 0.4931. Again precision, with 0.5487, was higher than recall, with 0.4477. Table 4 summarizes the F-score of this combined system compared to the F-score of just the CSD system by itself.

	F-score
0% WER, test set	0.8509
33.08% WER, test set	0.4931

Table 4: Performance of CSD system alone (0% WER) and performance of CSD system on ASR transcription (33.08% WER)

To put the combined systems' performance into context with results from the Airbus challenge, Figure 4 shows WER plotted against F-score for all teams with an entry for both systems, as well as the results obtained within this research.

Furthermore, 53.07% of true call signs were correctly transcribed by the ASR system, which means the CSD could at most have achieved a recall of 0.5307. The adjusted recall of the CSD is therefore:

$$\frac{0.4477}{0.5307} = 0.8436$$

4 Discussion

The study demonstrates that the ASR system of Hannun et al. (2019) performs quite poorly on the ATCC corpus (33.08% WER), compared to results reported in the paper (5.36-15.64% WER), and also below average compared to results of the Airbus challenge (mean 14.5% WER). The CSD system performs fairly well by itself (F-score 0.8509), but when paired with the ASR system the performance worsens (F-score 0.4931), and it is again below average compared to the Airbus challenge (mean F-score 0.6718). The results indicate that the performance of the call sign detection system is strongly affected by errors in the automatic

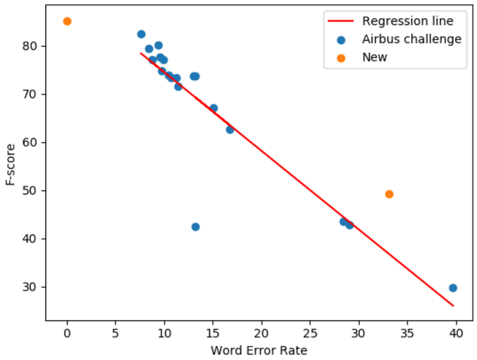


Figure 4: WER vs. F-score for all Airbus teams and new results obtained with Bi-LSTM and CRF architecture (both on human transcription - 0% WER and on ASR transcription - 33.08% WER), as well as a regression line fit to Airbus challenge results

speech recognition, seeing that there is a 0.35 drop in F-score from the CSD on the human transcription to the CSD on the ASR transcription. This, along with the findings that only 53.07% of true call signs were correctly transcribed by the ASR, suggests that the mediocre results of the combined systems are caused by errors in the ASR, to a higher extent than by errors in the CSD.

The poor performance of the ASR results compared to those in the paper of Hannun et al. (2019) is, to some degree, to be expected since ATC data has higher noise and accent variation than the data used in the original paper, making it more difficult to achieve low WERs. But moreover, the ASR results are also worse than those reported by Pellegrini et al. (2019) in the Airbus challenge, which used ATC data. This might be explained by the fact that the data used in the Airbus challenge was of higher quality, but this remains difficult to state since the corpus used there is not publicly available. Another plausible explanation for this difference in accuracy is that the Airbus corpus' size is two times that of the ATCC corpus.

As for the call sign detection, we see that there is a small drop in accuracy when using randomly generated call signs (with known telephony designators), which indicates that the system indeed benefitted from some call signs being present in both training and test data. But with an F-score of 0.8133, the system still performs well and overall seems to have learned the general structure of call signs. When using unknown telephony designators, the F-score drops to 0.6951, which suggests that the system more strongly relies on having encountered telephony designators in order to recognize a call sign. This is also the test case with the biggest discrepancy between precision and recall, which indicates that the CSD system only predicts few call signs, missing a lot of true call signs, but out of those predicted call signs, most are true call signs.

Although the performance of the combined ASR and CSD is below the average of the Airbus challenge in regards to F-score, the system does better than some Airbus teams that had lower WERs, as can be seen in Figure 4. This could mean either that the types of errors made by the ASR system are different to those made by systems in the Airbus challenge, and not impacting the CSD system's performance as severely, or that the CSD system used here is better at dealing with and overcoming the errors made by the ASR system.

Another interesting aspect is that when adjusting the recall of the combined system for the ASR system's errors, the value is higher (0.8436) than the recall of the CSD system tested on its own (0.8011). So even though overall accuracy is lower in combination, the CSD system itself is performing better in terms of recall. This is likely due to the set of 'available' true call signs simply being smaller in the adjusted case, but it shows again that the CSD system is performing well and that the ASR system's errors are the main cause of the poor performance of the combined systems.

The effect of errors in the ASR system on the CSD systems performance was to be expected since the CSD system's performance is evaluated against the true call signs in the human transcribed data but can only be as good as the ASR transcription allows it to be. In terms of recall, as already mentioned in section 2.5, the CSD system can at most detect as many call signs as were correctly transcribed by the ASR system, and hence we saw that the low recall of 0.4477 was largely due to the ASR system only transcribing 0.5307 of call signs correctly. The effect on precision is more difficult to adjust for, but is explained by the fact that call signs that are transcribed incorrectly by the ASR, while still keeping the structure of a call sign, for example, 'LUFTHANSA 787' wrongly transcribed as 'LUFTHANSA 797', are identified as call signs by the CSD, but counted as wrong since they are no exact match to the true call sign. This causes the CSD system's precision to be low, even though the system is good at recognizing the general structure of call signs.

These results should be taken into account when considering how to tune future call sign detection systems that are combined with an automatic speech recognition system. With the two aspects in mind, that when on their own, ASR does poorly and CSD does well and that when in combination, it's mostly errors in the ASR system that reduce the performance of the overall system, it is suggested that the priority should primarily be on improving and reducing errors in the ASR system.

4.1 Limitations

Due to the limited amount of data, of 20h of speech in the ATCC corpus, the results cannot confirm that the ASR system of Hannun et al. (2019) would perform poorly on ATC data in general. Additionally, the CSD system's performance was limited by the amount of data that had call sign annotations, so it is possible that higher F-scores could have been achieved on the ATCC data set with the full corpus available.

Another aspect that can be considered a limitation is the fact that both systems are trained and tuned individually, although the goal is to use them in combination. In research concerning named entity recognition in combination with automatic speech recognition, it has been suggested that training and tuning both together as one system could be a valuable option (Ghannay et al., 2018). Another suggestion from research on NER and ASR is to use evaluation metrics other than WER for tuning the ASR system (Jannet et al., 2015, 2017).

5 Conclusion

This research aimed to investigate how pre-existing systems of automatic speech recognition and call sign detection, by Hannun et al. (2019) and Gupta et al. (2019), perform on the *Air Traffic Control Communication* speech corpus, and further, how the errors of either system affect the final accuracy when the systems are applied in sequence. The research showed that the automatic speech recognition of Hannun et al. (2019) does not perform well, but the call sign detection system of Gupta et al. (2019) achieves fairly high accuracy, and in combination, they perform quite poorly again. The results further demonstrate that the combined systems' performance is strongly influenced by transcription errors of the automatic speech recognition system and that this seems to be the main cause for the low accuracy.

Before this type of combined automatic speech recognition and call sign detection can be applied to real-life air traffic control scenarios and become a safe and productive aid to pilots and controllers, it needs to be significantly improved in accuracy.

5.1 Future Research

Future research could explore whether training automatic speech recognition and call sign detection as one, rather than individually, could improve the performance accuracy of the system. Also, it could be interesting to see whether using other evaluation metrics when tuning the ASR system could improve the overall performance, as has been researched with NER.

Further research could also investigate if adding more data from different ATC corpora in the training of the ASR model could improve accuracy or if different ASR systems that were used on other ATC corpora perform better on the ATCC corpus.

Furthermore, it would be interesting to see more studies on the nature of the relationship between automatic speech recognition and call sign detection performance and whether they interact linearly or otherwise.

References

Delpech, E., Laignelet, M., Pimm, C., Raynal, C., Trzos, M., Arnold, A., and Pronto, D. (2018). A Real-life, French-accented Corpus of Air Traffic Control Communications. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 2866–2870, Miyazaki, Japan. European Language Resources Association (ELRA). ISBN: 979-10-95546-00-9.

Ghannay, S., Caubriere, A., Estève, Y., Camelin, N., Simonnet, E., Laurent, A., and Morin, E. (2018). End-To-End Named Entity And Semantic Concept Extraction From Speech. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 692–699, Athens, Greece. DOI: 10.1109/SLT.2018.8639513.

Gupta, V., Rebout, L., Boulianne, G., Ménard, P. A., and Alam, M. J. (2019). CRIM's Speech Transcription and Call Sign Detection System for the ATC Airbus Challenge Task. In *Proc. Interspeech 2019*, pages 3018–3022, Graz, Austria. DOI: 10.21437/Interspeech.2019-1131.

Hannun, A., Lee, A., Xu, Q., and Collobert, R. (2019). Sequence-to-Sequence Speech Recognition with Time-Depth Separable Convolutions. In *Proc. Interspeech 2019*, pages 3785–3789, Graz, Austria. DOI: 10.21437/Interspeech.2019-2460.

ICAO (2001). *ANNEX 10 to the Convention on International Civil Aviation, Volume II - Communication Procedures including those with PANS Status*, 6th edition.

ICAO (2017). *Designators for Aircraft Operating Agencies, Aeronautical Authorities and Services (Doc 8585/182)*, 182nd edition. ISBN: 978-92-9258-308-8.

Jannet, M. A. B., Galibert, O., Adda-Decker, M., and Rosset, S. (2015). How to Evaluate ASR Output for Named Entity Recognition? In *Proc. Interspeech 2015*, pages 1289–1293, Dresden, Germany.

Jannet, M. A. B., Galibert, O., Adda-Decker, M., and Rosset, S. (2017). Investigating the Effect of ASR tuning on Named Entity Recognition. In *Proc. Interspeech 2017*, pages 2486–2490, Stockholm, Sweden. DOI: 10.21437/Interspeech.2017-1482.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural Architectures for Named Entity Recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. DOI: 10.18653/v1/N16-1030.

Pellegrini, T., Farinas, J., Delpech, E., and Lancelot, F. (2019). The Airbus Air Traffic Control Speech Recognition 2018 Challenge: Towards ATC Automatic

Transcription and Call Sign Detection. In *Proc. Interspeech 2019*, pages 2993–2997, Graz, Austria. DOI: 10.21437/Interspeech.2019-1962.

Pratap, V., Hannun, A., Xu, Q., Cai, J., Kahn, J., Synnaeve, G., Liptchinsky, V., and Collobert, R. (2019). wav2letter++: The Fastest Open-source Speech Recognition System. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6460–6464, Brighton, United Kingdom. DOI: 10.1109/ICASSP.2019.8683535.

Rozenbroek, T. J. (2020). Sequence-to-Sequence Speech Recognition for Air Traffic Control Communication. Bachelor Thesis, Radboud University Nijmegen.

Sang, E. F. T. K. and Meulder, F. D. (2003). Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, page 142–147, Edmonton, Canada. DOI: 10.3115/1119176.1119195.

Saon, G., Kurata, G., Sercu, T., Audhkhasi, K., Thomas, S., Dimitriadis, D., Cui, X., Ramabhadran, B., Picheny, M., Lim, L.-L., Roomi, B., and Hall, P. (2017). English Conversational Telephone Speech Recognition by Humans and Machines. In *Proc. Interspeech 2017*, pages 132–136, Stockholm, Sweden. DOI: 10.21437/Interspeech.2017-405.

Šmídl, L. (2011). Air Traffic Control Communication. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. `http://hdl.handle.net/11858/00-097C-0000-0001-CCA1-0`.

Xiong, W., Droppo, J., Huang, X., Seide, F., Seltzer, M., Stolcke, A., Yu, D., and Zweig, G. (2017). Toward Human Parity in Conversational Speech Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25:2410 – 2423. DOI: 10.1109/TASLP.2017.2756440.

Appendices

A ASR Configuration

This appendix contains the configuration files used for the ASR system. The architecture of the network is contained in *network.arch*, the flags used for training are in *train.cfg*, and the flags used for decoding are in *decode.cfg*. All files are in wav2letter++ format.

A.1 Network Architecture

```
V -1 NFEAT 1 0
C2 1 12 25 1 2 1 -1 -1
R
DO 0.2
LN 0 1 2
TDS 12 25 80 0.2
TDS 12 25 80 0.2
C2 12 16 25 1 2 1 -1 -1
R
DO 0.2
LN 0 1 2
TDS 16 25 80 0.2
TDS 16 25 80 0.2
TDS 16 25 80 0.2
C2 16 20 25 1 2 1 -1 -1
R
DO 0.2
LN 0 1 2
TDS 20 25 80 0.2
TDS 20 25 80 0.2
TDS 20 25 80 0.2
TDS 20 25 80 0.2
TDS 20 25 80 0.2
TDS 20 25 80 0.2
V 0 1600 1 0
RO 1 0 3 2
L 1600 1024
```

network.arch

A.2 Training Flags

```
# Training config for ATC using Time-Depth Separable Convolutions
--runname=seq2seq_tds_tijs
--rundir=/home/s4841670/Documents/w2lmodel/runs
--tokensdir=/home/s4841670/Documents/w2ldata/am
--archdir=/home/s4841670/Documents/w2lmodel/configuration_files/
    seq2seq_tds_tijs
--datadir=/home/s4841670/Documents/w2ldata
--train=lists/atcctrain.lst
--valid=lists/atccvalid.lst
--lexicon=/home/s4841670/Documents/w2ldata/am/lexicon.txt
```

```
--arch=network.arch
--tokens=tokens.txt
--samplerate=8000
--criterion=seq2seq
--lr=0.05
--lrcrit=0.05
--momentum=0.0
--warmup=0
--stepsize=873045
--gamma=0.5
--maxgradnorm=15
--mfsc=true
--dataorder=output_spiral
--inputbinsize=25
--filterbanks=80
--attention=keyvalue
--encoderdim=512
--attnWindow=softPretrain
--softwstd=4
--trainWithWindow=true
--pretrainWindow=14551
--maxdecoderoutputlen=120
--usewordpiece=true
--wordseparator=_
--sampletarget=0.01
--target=ltr
--batchsize=4
--labelsmooth=0.05
--nthread=4
--memstepsize=4194304
--eostoken=true
--pcttraineval=1
--pctteacherforcing=99
--iter=4850250
--enable_distributed=true
```

train.cfg

A.3 Decoding Flags

```
# Decoding config for ATCC corpus using Seq2Seq TDS model
# for test-clean (best params for dev-clean)
--am=/home/s4841670/Documents/w2lmodel/runs/seq2seq_tds_tijs/001
    _model_lists#atccvalid.lst.bin
--lm=/home/s4841670/Documents/w2ldata/lm/atcc3gram.binary
--tokendir=/home/s4841670/Documents/w2ldata/am
--tokens=tokens.txt
--lexicon=/home/s4841670/Documents/w2ldata/am/lexicon.txt
--test=lists/atcctest.lst
--uselexicon=true
--sclite=/home/s4841670/Documents/w2lmodel/decode_logs/
    seq2seq_tds_tijs
--decodertype=tkn
--lmweight=1.0
--wordscore=2.0
--beamsize=80
```

```
--beamthreshold=7
--hardselection=1.5
--softselection=10.0
--smearing=max
--show=true
--showletters
--eosscore=-3
```

decode.cfg

B Unknown Telephony Designators

Table 5 shows the telephony designators that were used to generate new, unseen call signs.

DEVIL	FROSTY	WINGSPAN	GLOBETROTTER	ANDAX
TWISTER	ISLANDWAYS	SHAMROCK	VOLTA	COTSWOLD
FRACJET	ARROW	MULTISKY	AEROCOM	AMADEUS
PEARL	GEOLINE	ROUGE	LEGEND	GENEX
AEROCUTTER	LUNA	BOREALIS	AFRISPIRIT	ARABIA
AIRSPEC	AIRCALIN	LUXORJET	KAUNAS	ALBASTAR
MEKONG	PEGASUS	TWINGOOSE	INTEGRA	SAHARA
BANDAMA	SNOOPY	BOURBON	ATLANTA	TANZANIA

Table 5: Unknown telephony designators