Transportation Research Record
2021, Vol. XX(X) 1–12
©National Academy of Sciences:
Transportation Research Board 2021
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/ToBeAssigned
journals.sagepub.com/home/trr
SAGE

Automatic Speech Recognition for Air Traffic Control Communications

Sandeep Badrinath[1] and Hamsa Balakrishnan[2]

Abstract

A significant fraction of communications between air traffic controllers and pilots is through speech, via radio channels. Automatic transcription of air traffic control (ATC) communications has the potential to improve system safety, operational performance, conformance monitoring, and to enhance air traffic controller training. We present an automatic speech recognition model tailored to the ATC domain that can transcribe ATC voice to text. The transcribed text is used to extract operational information such as call-sign and runway number. The models are based on recent improvements in machine learning techniques for speech recognition and natural language processing. We evaluate the performance of the model on diverse datasets.

Introduction

Air traffic controllers play a critical role in ensuring safe separation of aircraft in the airspace as well as on the airport surface. Despite the increasing deployment of datalink technologies, a significant fraction of communications between air traffic controllers and pilots is through speech, via radio channels. The automatic transcription of air traffic control (ATC) communications has many potential applications such as improving system safety, operational performance, conformance monitoring, and enhancing controller training (*1*). However, automatic speech recognition (ASR) systems proposed to date for the ATC applications have not yet demonstrated the levels of accuracy needed for practical deployment (*2*).

Factors such as noisy radio channels, high speech rates, and diverse accents pose challenges to the development of ASR systems for ATC. On the other hand, ATC communications contain domain-specific vocabulary and standard phraseology that can be leveraged to tailor algorithms. Recent developments in machine learning such as deep neural networks have led to more accurate speech recognition algorithms (*3*), and suggest the possibility of better ASR algorithms for ATC communications.

Related work

ASR techniques have several potential applications in the ATC domain, including safety monitoring of live operations (*4*–*6*) and identification of anomalous aircraft trajectories (*4*). In another example of safety monitoring, Chen et al. (*5*) developed a framework to automatically flag pilot-controller miscommunications (read back error detection) to prevent untoward incidents. With the introduction of electronic flight strips in ATC towers, speech assistants have been shown to reduce air traffic controller workload in early demonstrations (*7*). ASR systems can also be used instead of pseudopilots for training air traffic controllers (*8*, *9*), and for human-in-the-loop simulations and workload measurements in air traffic management research (*10*, *11*). Additionally, the increasing demand for unmanned aerial vehicle operations has stimulated the need for ASR systems (*12*). However, the lack of sufficiently accurate ASR models for the ATC domain has remained a significant barrier to its deployment in these applications.

The earliest automatic speech recognition models for large vocabularies used Hidden Markov Models (HMM) (*13*). Later approaches used hybrid models – a mixture of HMMs with Gaussian Mixture Models (GMM) or Deep Neural Networks (DNN) (*14*, *15*). Recently, end-to-end speech recognition using deep neural networks have yielded significant improvements in accuracy for regular conversational speech (*3*). A key benefit of the end-to-end speech recognition model over classical approaches such as HMM-based models is the ease of training, since they do not require complicated pipelines with extensively engineered

[1] PhD Candidate, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, 02139.
[2] Professor, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, 02139.

Corresponding author:
Sandeep Badrinath, sandeepb@mit.edu

processing stages. There are several open source ASR toolkits that allow researchers to adapt various models to a domain of interest: Bavieca (16), CMU Sphinx (17), HMM toolkit (18) and Kaldi (19) for HMM-based ASR models, and Deep Speech (20), wav2letter++ (21) and Jasper (22) for deep-learning based end-to-end speech recognition models.

ATC applications of ASR have traditionally used HMM-based approaches, with limited consideration of end-end speech recognition approaches (6). In the recently-concluded Airbus ATC speech recognition challenge, the top performing team used a HMM-based hybrid model that yielded a word error rate (WER) of 7.6% on the Airbus test data (23), with the second ranking team using an end-to-end speech recognition architecture that yielded a WER of 8.4% (24). In another recent study, a comparison of hybrid model (HMM/DNN in Kaldi) and end-to-end speech recognition for ATC voice in English and Chinese was evaluated by Lin et al. (6). For English ATC voice, they found that the end-to-end speech recognition model yielded a WER of 6%, performing better than the hybrid model that yielded a WER of 9%. One needs to note that the performance of the models, in terms of the WER, also depends on the quality of test data. If the test data is more representative of the training data and is less noisy, one expects better performance from the speech recognition model. Therefore, one cannot compare models based on WER if they are reported on different test sets.

The accuracy of a speech recognition model depends on the amount of labelled training data available. The amount of publicly available transcribed ATC voice data is relatively small compared to thousands of hours of transcriptions available for regular conversational speech (used for training commercial speech recognition models). To tackle the issue of limited data availability, Srinivasamurthy et al. (25) proposed using semi-supervised learning, in which new transcripts generated from a preliminary trained model were used to retrain the model, resulting in a 25% reduction in the word error rate. Researchers have also used flight trajectories in order to add context to a speech recognition model (for example, flagging a flight as about to land) to improve its accuracy. Such a context-aware ASR model has shown to reduce the command error rate (a measure of extracting commands from the transcripts) by 50%, although there were no significant improvements in the WER (26). Prior work has also leveraged the smaller vocabulary and standard phraseology of ATC communications in order to develop better language models, thereby achieving up to 20% improvement in the accuracy of the speech recognition model (27).

Contributions of this paper

In this paper, we develop an automatic speech recognition (ASR) model that transcribes ATC voice communications to text. The proposed model is based on an end-to-end speech recognition architecture with a deep neural network, which offers several advantages over traditional HMM-based approaches that have been typically used for the ATC domain. HMM-based models typically contain multiple modules (acoustic model, pronunciation model, etc.), with each module being independently optimized with their own objective function which does not guarantee optimality across modules (28). By contrast, an end-to-end model replaces multiple modules with a single deep neural network that enables direct mapping of the acoustic signals to a sequence of characters, without hand-engineered intermediate states. As a result, one could attain optimality over the entire pipeline with end-to-end models by merging multiple modules into one optimized deep neural network, and designing objective functions that truly reflect the final evaluation criterion. Moreover, an end-to-end modeling approach is easier to train, and has been shown to yield a better accuracy than traditional methods for conversational speech (3). In addition to training the model with ATC voice transcripts, we compare the model accuracy with transfer learning and parameter fine tuning using a model that is pre-trained on regular conversational speech. Since the accuracy of the speech recognition model depends significantly on the amount of training data, we have compiled an extensive ATC speech corpus from various sources to understand the impact of using diverse ATC voice datasets for training the model.

Most applications require the extraction of information from the transcripts once the ATC voice data has been transcribed. For this purpose, we present a methodology to accurately extract operational information such as aircraft call signs and runway assignments from transcripts, using state-of-the-art techniques from natural language processing. We also propose a performance metric called normalized uncertainty score to evaluate the accuracy of transcription in the absence of ground truth, a necessity for making decisions using transcribed ATC voice data.

Automatic speech recognition model

The automatic speech recognition model is based on Deep Speech (20), an end-to-end speech recognition model. We use Mozilla's implementation of Deep Speech for our analysis (29). We briefly describe the model in this section; more detailed information can be found in the original papers (20, 29).

Model overview

Figure 1 illustrates the model architecture of the speech recognition system. The main components are the feature extraction module, acoustic model, language model, and decoding module. The feature extraction module takes the ATC audio signal as input, and outputs coefficients associated with its frequency spectrum as follows: the entire time series of the audio signal is divided into smaller (32 ms) time-windows with a 20 ms overlap; each time-window is

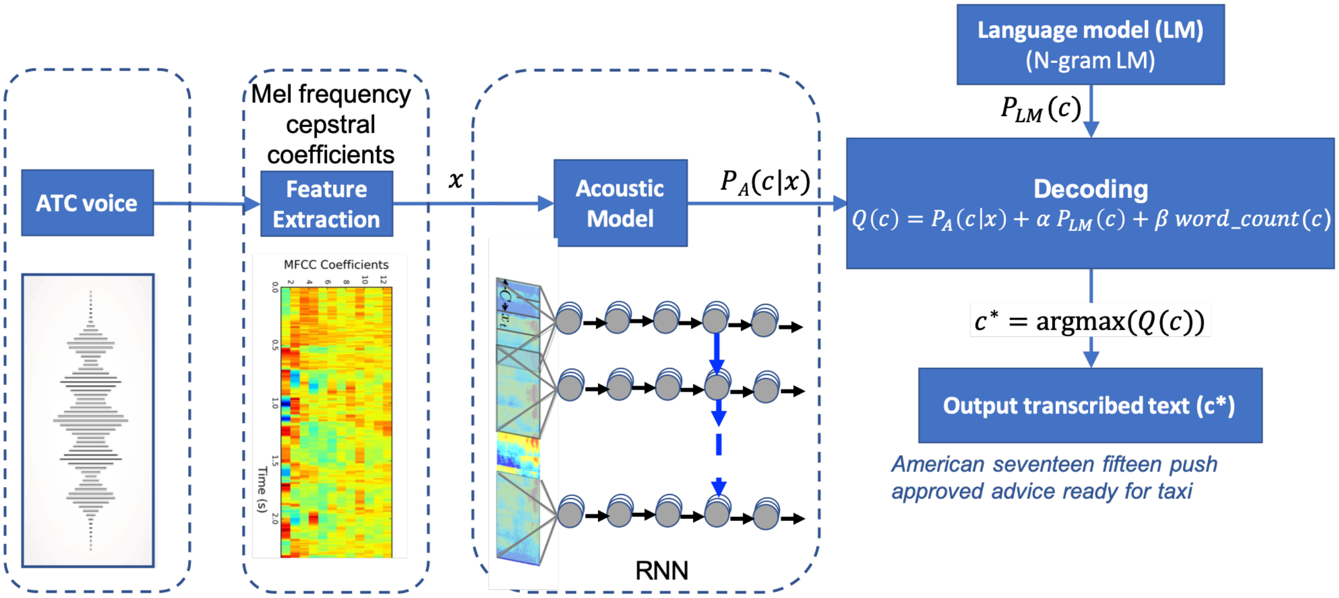

Language model (LM)
(N-gram LM)
$P_{LM}(c)$
Mel frequency cepstral coefficients
ATC voice
Feature Extraction
x
Acoustic Model
$P_A(c|x)$
Decoding
$Q(c) = P_A(c|x) + \alpha\, P_{LM}(c) + \beta\, word\_count(c)$
MFCC Coefficients
Time (s)
RNN
$c^* = \text{argmax}(Q(c))$
Output transcribed text (c*)
American seventeen fifteen push approved advice ready for taxi

Figure 1. Model architecture for the ASR system.

associated with a feature vector that corresponds to its Mel Frequency Cepstral Coefficients (MFCCs) (*29*). The MFCCs of each time-window constitute the feature vector that serves as the input to the acoustic model. The acoustic model is a recurrent neural network that is trained to output a sequence of character probabilities based on the sequence of input feature vectors. The characters here correspond to letters of the English alphabet, apostrophe, space, blank, and additional identifiers corresponding to foreign words or unintelligible words. The language model outputs the probability of a sequence of characters based on the training text data, independent of the audio signal. In the decoding module, the output of the acoustic model is integrated with that of the language model to determine the transcribed text.

Acoustic model

The acoustic model is a recurrent neural network composed of five hidden layers, the first three of which are non-recurrent. The first layer takes the MFCC input for a particular time-window as well as a few (9, in our case) context frames on either side of the window. A clipped rectified-linear unit (ReLU) activation function is used for all the layers, except the last one. The model architecture of Mozilla's implementation differs slightly from the original Deep Speech paper (*20, 29*). Here, the fourth layer is a feed forward recurrent layer instead of a bidirectional recurrent layer, to reduce computational time during inference. The last layer is a non-recurrent output layer that yields the character probabilities based on a Softmax function. A Connectionist Temporal Classification (CTC) loss function is used to compute the prediction error during training (*30*). We use an adaptive learning rate (Adam method) for updating the model parameters through stochastic gradient descent during training, and a dropout rate of 5% for regularization.

Language model

Training the acoustic model for the ATC domain to produce accurate character level transcription is challenging due to the limited availability of transcribed ATC audio. Furthermore, the neural network can output phonetically similar word renderings which can be incorrect (e.g., ”bostin” instead of ”boston”). These issues can be addressed using a language model that is tailored to the ATC domain. The language model is a probability distribution over a sequence of words, which is used in the decoding stage. We choose an N-gram language model because it can be trained using existing libraries (KenLM (*31*)) and yields good performance. In this language model, the probability of the k^{th} word is assumed to depend only on the $N-1$ preceding words, i.e., $P(w_k|w_{k-1}, w_{k-2}, .., w_1) = P(w_k|w_{k-1}, w_{k-2}, .., w_{k-(N-1)})$. Consequently, the probability of a sequence of words, $P(w_1, w_2, ...w_m)$, can be expressed as a product of conditional probabilities.

The conditional probabilities required for the model are determined from the ATC audio transcripts in the training data. Although a higher value of N will lead to better predictions, there may not be sufficient data to obtain consistent statistics for a larger N. An optimal value of N is determined using a parametric analysis to yield a lower word error rate on the validation set.

Decoding step

The decoding step determines the most probable sequence of characters given the output probabilities from the acoustic model and language model. Let $P_A(\mathbf{c}|\mathbf{x})$ represent the probability of a sequence of characters, $\mathbf{c} = \{c_1, c_2, c_3..\}$, obtained from the acoustic model for a given input audio ($\mathbf{x}$). Similarly, let $P_{LM}(\mathbf{c})$ represent the probability of a sequence of characters obtained from the language model. The objective of the decoding step is to obtain a sequence of characters that maximizes the confidence score ($Q(\mathbf{c})$):

$$Q(\mathbf{c}) = P_A(\mathbf{c}|\mathbf{x}) + \alpha P_{LM}(\mathbf{c}) + \beta \text{ word count}(\mathbf{c}). \qquad (1)$$

Here, α and β are weights to balance the influence of the acoustic model, the language model, and the word count of the utterance. The last term is used because shorter sentences inherently have a higher probability in the N-gram language model. The output sequence of characters that maximizes the confidence score, $Q()$, is determined using a beam search algorithm (20). Optimal values of α, β and beam width are determined using a parametric analysis such that the average WER over a validation set is minimized.

Model for ATC communications

Data sources

Training an automatic speech recognition model requires large amounts of transcribed audio data. The datasets used in our study are shown in Table 1, and include transcripts of ATC communications from the US and Europe, with varied accents and audio quality. Most of the data corresponds to approach and tower control segments. The European datasets also contain utterances that are partially non-English. Those parts, as well as non-intelligible parts of the audio, are encoded with a unique identifier in the transcripts. The total number of hours of audio transcription is approximately 140 hours, yielding 84 hours after removing outliers and silent portions of audio. To the best of our knowledge, this is the first study that combines these four diverse datasets.

Performance metrics

The Word Error Rate (WER) is a widely-used measure of ASR model accuracy. It is defined as the sum of the number of words in the transcribed text that are either substituted (S), deleted (D), or inserted (I) relative to the reference text, divided by the number of words in the reference (N_r):

$$WER = \frac{S + D + I}{N_r}. \qquad (2)$$

The WER includes errors arising from filler words or other words that are not relevant in a particular context. The Concept-Error-Rate (CER) is an alternative metric that reflects the accuracy of domain-specific ASR systems. The CER is given by number of misrecognized concepts divided by the total number of concepts. This study focuses on the WER primarily because the number of concepts in the speech that one would be interested in (for evaluating the CER) depends on the application area. Additionally, one needs the labels of the concepts in the training data to evaluate the CER, which is not available in our dataset.

The speed of transcription is also an important quantity. It is measured in terms of the real-time factor (RTF), namely, the duration of input audio divided by the required time to process the input. Lower RTF, which implies faster transcription speed, is preferred. However, RTF might not be significantly important for off-line applications that involve accurately transcribing recorded voice datasets. On the other hand, if one is interested in transcribing the speech in real time (in applications such as safety monitoring), then one needs to have RTF lesser than one. The real-time factor for our models is around 0.3, implying that they can be deployed in live operations.

Model Variants

We consider the following four model variants in our study:

Model-1: Only an acoustic model and no language model Here, the transcription is based only on the acoustic model, and the language model weights (α, β) are set to zero. This variant serves as a baseline to assess the benefit of including a language model. The number of neurons in each hidden layer of the neural network was fine-tuned based on a parametric analysis that involved varying its value in the range (500,1024) and picking the optimal value that yielded the lowest average WER on the validation set. The optimal number was found to be 650 neurons in each hidden layer. We note that the original Deep Speech model (20) had a significantly higher number of neurons in each hidden layer (2,048 neurons). However, we found that a fewer number of neurons was optimal because of the smaller amount of training data available relative to conversational speech. In addition to English characters, some of the datasets use special characters to represent foreign words (@), non-intelligible words (_), breaks (/) and noise (#). We include these special characters in our training data.

Model-2: Acoustic model and language model The language model weights (α, β) are chosen by sweeping through different values of (α, β), and choosing the set of parameters that yields the lowest WER over a validation set. Figure 2 shows the computed WER on the validation set for different values of (α, β). Using this, we determined the parameters that yield the lowest WER. A similar parametric analysis resulted in a choice of $N = 4$ for the N-gram language model. Ideally, one should search for the optimal set of parameters over the joint-space of the hyperparameters. However, to limit the number of computations, we individually optimized the beam width, N-value of the language model and the number of neurons,

Table 1. Available transcribed ATC voice datasets

Corpus name	Duration	Comments
AIRBUS-ATC (*23*)	40 hours	French-accent speech; approach and tower control, ATIS
ATCOSIM Corpus (*32*)	10 hours	German-Swiss-French accent; en-route controllers speech recorded in a studio environment
ATC Communication (*33*)	20 hours	Ground control, tower control, approach control, and area control. Primary data source from Czech Republic's ANSP, with small amounts from Lithuania and Philippines.
NIST ATC Complete (LDC94S14A) (*34*)	70 hours	ATC data for approach control from 3 US airports (BOS, DCA, DFW)

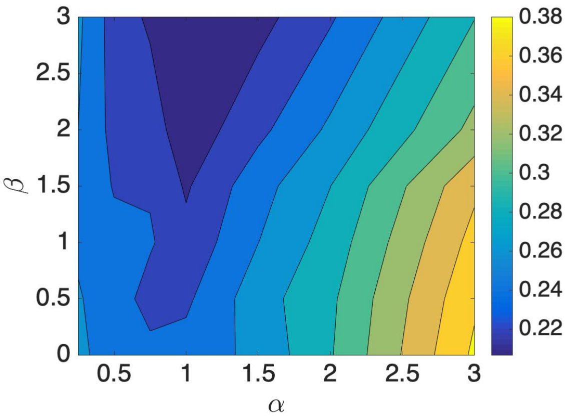


Figure 2. WER of validation set for various language model weights (α, β).

while keeping the other parameters fixed at their nominal values.

Model-3: Refining parameters of a pre-trained model We used an acoustic model that was trained for regular conversational speech, and refined the neural network parameters using the ATC speech corpus. The pre-trained model is a much larger neural network with 2,048 neurons within each layer. The pre-trained acoustic model was developed using multiple speech corpora (3,250 hours with Fisher, LibriSpeech and Switchboard datasets) and achieves a WER of 0.08 on conversational speech (LibriSpeech clean test corpus) (*29*). The neural network parameters of the pre-trained model were optimized by re-training the model using the ATC speech corpus. The optimization was conducted over four epochs using a small learning rate (10^{-5}) for the gradient descent. The choice of four epochs was based on common practice, and can be further revised considering the performance over the validation set. As before, we used a language model that was trained using the ATC corpus.

Model-4: Transfer learning with a pre-trained model Transfer learning has shown promise in machine learning applications where there is limited domain-specific labelled training data available (*35*). The idea is to adapt a pre-existing trained model to the domain of interest. In our case, we adapted a pre-existing acoustic model that is trained for regular conversational speech (the same baseline model used for Model-3) to the case of ATC communications. The most common approach to transfer learning with neural networks is to ”freeze” the parameter values of a few layers of a trained neural network, and to retrain the parameters of the other layers using domain-specific data. For ASR, the parameters of the first few layers are frozen from the pre-trained model, and parameters of the remaining layers are trained using the ATC speech corpus. The rationale for such an approach is that the first few layers represent some form of filtering or feature extraction that might be common across different application domains. This methodology has been used to train ASR models for different languages using a pre-trained model for the English language (*36*). In our study, we froze the parameters of the first three layers of a pre-trained ASR model for conversational speech (from the previous variant), and retrained the parameters of the last two layers using the ATC speech corpus.

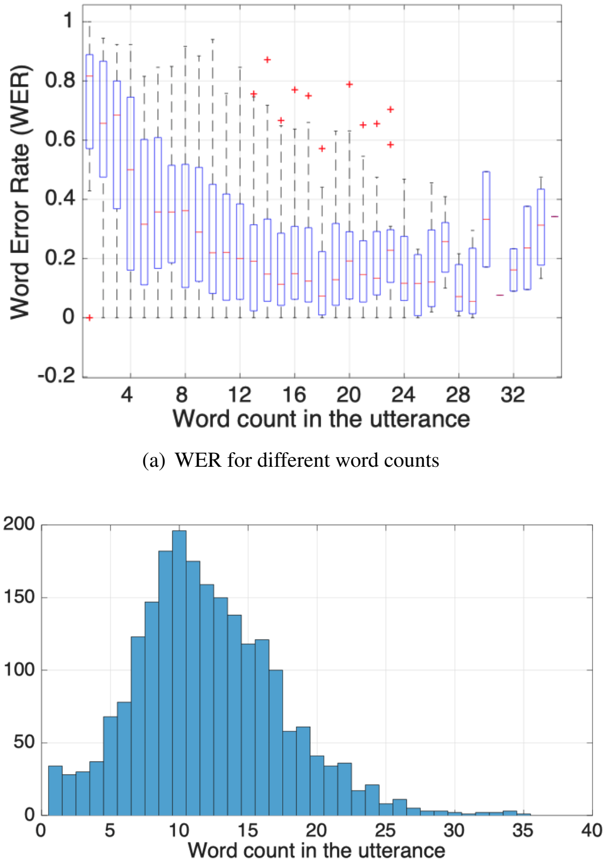
Model performance

We first trained and tested the different model variants with just one of the datasets (AIRBUS-ATC), and then included the other datasets. The initial analysis with a single dataset allowed us to test different model variants without significant computational effort. We used approximately 36 hours of the 40-hour AIRBUS-ATC dataset for training the model, 2 hours for validation, and 2 hours for testing.

A summary of the WERs of the four models (on a 2-hour test dataset with 1,500 utterances) is shown in Table 2. The acoustic model integrated with the language model (Model-2) performs the best among the four approaches with an average word error rate of 0.22. This is followed by the model obtained by fine-tuning a pre-trained ASR model (Model-3). By comparing results from Model-2 and Model-1, we notice that integrating the language model with the acoustic model improves the speech recognition accuracy by about 18%. Transfer learning yields the lowest accuracy among the different approaches, possibly because the higher speech rates and noise of ATC audio require significantly different parameter values in the first few layers of the neural network compared to a model trained on conversational speech.

Figure 3(a) shows a box-plot of the average WER as a function of the word count in the utterance. The WER is seen to be higher for utterances with a lower word count, and is nearly constant for word counts greater than six. The reason for this behavior is that missing one or two words (even filler words, e.g., ‘um’ or ‘ah’) would lead to a higher WER for smaller sentences. Figure 3(b) shows the histogram of the word count of utterances in the test set. The average word count is 12 words, and utterances with fewer than six words represent 7% of the test set. Additionally, 10.3% of the utterances in the test set have a foreign word or non-intelligible word in the transcription. We therefore compute the WER after excluding utterances with fewer than six words and the ones having foreign words or non-intelligible words, and denote it by $\overline{WER}$. One needs to note that shorter sentences might be important depending on the application (such as detecting acknowledgements or readback errors). For applications such as extracting operational information, the newly defined performance metric, $\overline{WER}$, might be of interest because such information is primarily contained in longer sentences. The $\overline{WER}$ of the four model variants is shown in Table 2. We note that $\overline{WER}$ is 8-25% lower than the WER. Overall, Model-2, wherein the acoustic and language models are trained using only ATC domain-specific speech data, performs the best among these approaches, with $\overline{WER}$ equal to 0.17.

As one might expect, it takes more time to train a neural network model from scratch (Models-1,2) than it takes for parameter fine tuning (Model-3), where the optimization is conducted over a smaller number of epochs (four in our case), or even transfer learning (Model-4) where only the last two layers of the neutral network are re-trained. However, we use fewer neurons for training the model from scratch compared to the pre-trained model, which slightly reduces the computational time (however, it is still higher than the other two methods).



(a) WER for different word counts

(b) Histogram of word count

Figure 3. Statistics of word counts and WERs computed for the test set using Model-2.

Training with additional datasets

To understand the benefit of additional training data, we added utterances from the three other corpora (ATCOSIM Corpus, ATC Communication, and NIST ATC Complete) to the AIRBUS-ATC data, and re-trained the acoustic model. The training set comprised of about 65,500 utterances (80 hrs of non-silenced audio) of ATC conversations. A comparison between the WER computed using the newly trained model with additional data and the earlier model, evaluated on the AIRBUS-ATC and NIST ATC Complete (100 utterances) testsets, is shown in Table 3. The $\overline{WER}$ on the AIRBUS-ATC test set increases from 0.17 to 0.29 with additional data. A possible reason for this increase is the fact that the datasets are diverse with regards to the accents, audio quality, and navigational information. However, $\overline{WER}$ evaluated on the NIST ATC complete dataset reduces from 0.75 to 0.32, illustrating that training with additional data increases the generalizability of the model. These comparisons use the same language model (trained using AIRBUS-ATC data) even for the evaluation with the NIST ATC data, which contributes to the lower accuracy on that test set. We also retrained the model that was initially trained with all the four datasets by parameter fine-tuning with the AIRBUS-ATC training set (similar to the approach explained in the previous section). The results indicate that parameter fine-tuning reduces the WER on the AIRBUS-ATC test set (Table 3) compared to case without fine-tuning, but still yields lower accuracy than when the sources of the training and test sets are the same (i.e., AIRBUS-ATC).

Extracting operational information

ATC communications contain extensive operational information (e.g., runway assignment, heading, flight level) that are of value in decision-making. In this section, we present a methodology to automatically extract operational information from ATC communication transcripts. Natural language processing (NLP) techniques for extracting information from unstructured text can be broadly classified into statistical

Table 2. WERs and $\overline{\textbf{WER}}$s on the AIRBUS-ATC test set for different model variants.

Variant	Model type	WER	$\overline{\text{WER}}$
Model-1	Acoustic model without language model	0.27	0.23
Model-2	Acoustic model integrated with language model	0.22	0.17
Model-3	Parameter fine-tuning with pre-trained model	0.25	0.23
Model-4	Transfer learning with pre-trained model	0.54	0.52

Table 3. Effects of using additional training data sources.

Training data source	$\overline{\text{WER}}$ on test set	
	AIRBUS-ATC	NIST ATC Complete
AIRBUS-ATC training set	0.17	0.75
All data sources	0.29	0.32
All data sources + fine-tuning with AIRBUS-ATC training set	0.19	0.45

models and rule-based grammar. Statistical models are obtained using machine learning algorithms on the text data. On the other hand, rule-based grammar techniques are based on a collection of hand-engineered rules to perform the NLP task. We employ both the methodologies in this paper: a statistical model is used for call-sign extraction, and a rule-based grammar approach is used to extract runway information.

Call-sign extraction

The utterances of the air traffic controllers and the pilots often contain the aircraft call-sign, which serves as a unique identifier for a flight receiving or sending the information. The call-sign typically comprises of the airline telephony designation (such as ”American” for ”American Airlines”; ”Speedbird” for ”British Airways”), followed by the flight ID, which comprises of 2-4 digits, with an optional suffix of 2-3 letters. A few examples of ATC utterances are shown below, with the aircraft call-sign highlighted:

- **air nostrum eight seven six one** contact bordeaux one three three decimal seven seven five goodbye
- five thousand feet one zero one zero **easy six eight four romeo**
- **lufthansa zero one charlie** reduce speed two two zero knots

To associate a flight with the instruction or command given by the air traffic controller or pilot, one must be able to accurately extract call-signs from the transcript. There are multiple challenges to accurate call-sign extraction: (a) controllers could use multiple airline identifiers for the same airline; (b) the number of words following the airline identifier (corresponding to the numerals/NATO phonetics) can vary; and (c) there could be call-signs with just the flight number without the airline identifier. Although the digits are expected to be pronounced as separate numbers as per ICAO nomenclature, the convention can vary in different regions (for example, in the US, ”2020” is often spoken as ”twenty twenty” instead of ”two zero two zero”). One could also have non-standard pronunciation: for example, “0” could be pronounced as ‘oh’, “9” could be pronounced as ‘nine’ instead of ‘niner’, and “F” could be pronounced as ‘fox’ instead of ‘foxtrot’.

Given these variations, a rule-based grammar approach for call-sign extraction might not be feasible. We instead use a statistical approach to train a model to identify call-signs using transcripts of ATC communications. In order to extract call-signs, we use Named Entity Recognition (NER), a standard NLP technique, to classify information within unstructured text into predefined categories. For the call-sign extraction problem, the objective is to identify a sequence of words in each utterance, and to categorize it as the call-sign, if it exists. We use a Python library called Spacy, which is one of many standard libraries for NER (*37*). The NER model in Spacy is based on a deep convolution neural network and uses sub-word features. To categorize a particular word, the model accounts for the neighbouring three words on either side. A key advantage of the sub-word feature is that it can identify call-signs even if there are spelling errors in the transcripts, a particularly useful property for ASR-generated transcripts.

Performance metrics The accuracy of the model is evaluated using three performance measures: precision, recall, and F-1 score. We introduce some notation to define these performance measures: Let A be the number of instances in the test set when there is no call-sign in the reference text, and the model also detects no call-sign. Similarly, let B be the number of instances when there is a call-sign in the reference text, but the model does not detect a call-sign. Let C be the number of instances when there is no call-sign in the reference text, but the model erroneously detects a call-sign. Let D and E be the number of instances when the detected call-sign is correct and incorrect, respectively, among the instances when the reference text has a call-sign. The mathematical expressions for precision (P), recall (R), and F-1 score are as follows:

$$P = \frac{D}{C+D+E}; \quad R = \frac{D}{B+D+E}; \quad F-1 = \frac{2PR}{P+R}. \tag{3}$$

Precision is the fraction of instances for which the call-sign is correctly extracted from among the instances when a call-sign is extracted. Recall is the fraction of instances for which the correct call-sign is extracted from among the instances when the call-sign is present in the reference text. The F-1 score is the harmonic mean of precision and recall.

Results The model was trained using 25,000 samples of labelled data, which contained the transcript of the utterance, and the corresponding call-sign in the utterance (if present). The independent test data set contained approximately 3,000 utterances. For the call-sign extraction task from the reference transcript, the precision was 0.97, the recall was 0.93, and the F-1 score was 0.95. In other words, 97% of the call-signs that were extracted matched the reference call-sign, and 93% of the call-signs in the reference text were extracted correctly. Table 4 shows a few example transcripts of the utterances in the test set, along with the reference call-sign contained within the transcript, and the extracted call-sign from the model. The first three rows in the table are instances in which the call-sign was correctly extracted. The next three rows illustrate cases wherein the call-sign present in the text was not extracted, which would impact the recall score. In these cases, the primary reason for the error was that either the airline identifier in the test set was very different from those seen in the training set, or that the call-sign did not follow the usual convention. The last row in the table shows a case in which the model incorrectly extracted a call-sign when there was none in the reference transcript, which would impact the precision score. Overall, a F-1 score of 0.95 is a good initial step, and can be improved by increasing the amount of training data. One needs to note that even if the call-sign was not seen in the training data, the model is capable of correctly identifying the call-sign based on neighboring words that gives it context. Table 5 shows the performance measures for the call-sign extraction task on the test set for different amounts of data in the training set. The performance of the model improves as the amount of training data increases, but with diminishing marginal returns, as one would expect.

Next, we look at the performance of the call-sign extraction task on the transcribed voice data that is output from the speech recognition model. These results are based on the transcripts generated from an ASR model with a $\overline{WER}$ of 0.17. For the call-sign extraction task from the transcribed voice data, the precision is 0.81, recall is 0.57, and the F-1 score is 0.67, on an independent test set. The performance here is worse than in the case where we had the actual transcripts, because of inaccuracies in the voice transcription. It is worth noting that even when the transcription of voice is not completely accurate (as reflected by the word error rate), it may be possible to accurately extract the call-sign from the transcript if the call-sign part of the utterance is transcribed correctly.

Some limitations of this statistical approach to call-sign extraction must be noted: (a) detecting airlines unseen in the training set can be challenging, even though the approach considers neighboring words for context; (b) call-sign digits are pronounced differently in the US and in Europe. These limitations can be overcome by including more diverse data sources for training.

Extracting runway information

A rule-based grammar was used for extracting runway information from the transcript. Reasons for employing a rule-based grammar approach rather than a statistical approach for runway extraction include: (a) the datasets do not contain the labelled runway information needed to build a statistical model; and (b) relatively few runway numbers are uttered in the datasets (because the voice recordings are from a small number of airports), which is not sufficient to obtain a statistical model that generalizes well. Furthermore, the utterances of runway numbers are highly structured, motivating the use of rule-based grammar. Runway numbers are uttered in the following manner: one or two digits followed by ‘left’, ‘right’ or ‘center’. The transcript is searched for such patterns to obtain the runway information for extraction. We hand-labelled about 200 utterances to test the performance of a rule-based grammar approach for extracting runway information. The precision, recall and F-1 score is 1 on the reference transcript, indicating that the rule-based approach is perfectly accurate in extracting runway information. Runway information extraction using the transcribed text (from the ASR model with $\overline{WER}$ of 0.17) shows very good performance, achieving a precision of 0.97, recall of 0.93 and F-1 score of 0.95. Note that for airports with a single runway (which were not present in the data used for this analysis), runway numbers might not include ‘left’, ‘right’, or ‘center’ modifiers. In such cases, one needs to appropriately modify the rule-based grammar by including ’runway’ followed by a digit as a keyword identifier.

Extensions and opportunities for further research

In this section, we briefly discuss some promising extensions and directions for further investigation.

Evaluating transcription accuracy in the absence of ground truth data

Calculation of the word error rate requires the availability of ground truth data (i.e., the actual transcript). However, the practical deployment of decision-making using ASR requires an estimate of the likely accuracy of the transcription, even in the absence of ground truth data. In other words, it

Table 4. Extracting call-signs from reference transcripts.

Reference call-sign	Reference Transcript	Extracted call-sign
easy four seven tango lima	roger we call you huh short final easy four seven tango lima	easy four seven tango lima
beeline four papa golf	report huh short final maximum one sixty beeline four papa golf	beeline four papa golf
swiss one juliet bravo	keep rolling and vacating via mike eight swiss one juliet bravo	swiss one juliet bravo
easy _ whiskey	toulouse tower @ easy _ whiskey established on the ils three two right	
airbus delta sierra	airbus delta sierra after takeoff you will maintain runway axis	
binair seven alpha	# huh good huh @ huh binair seven alpha approaching november eight	
	we leave via sierra three huh india lima	sierra three huh india lima

Table 5. Call-sign extraction performance for different amounts of training data.

Training dataset size (# of utterances)	Precision	Recall	F-1 score
100	0.91	0.34	0.50
500	0.95	0.49	0.65
5,000	0.99	0.90	0.94
25,000	0.97	0.93	0.95

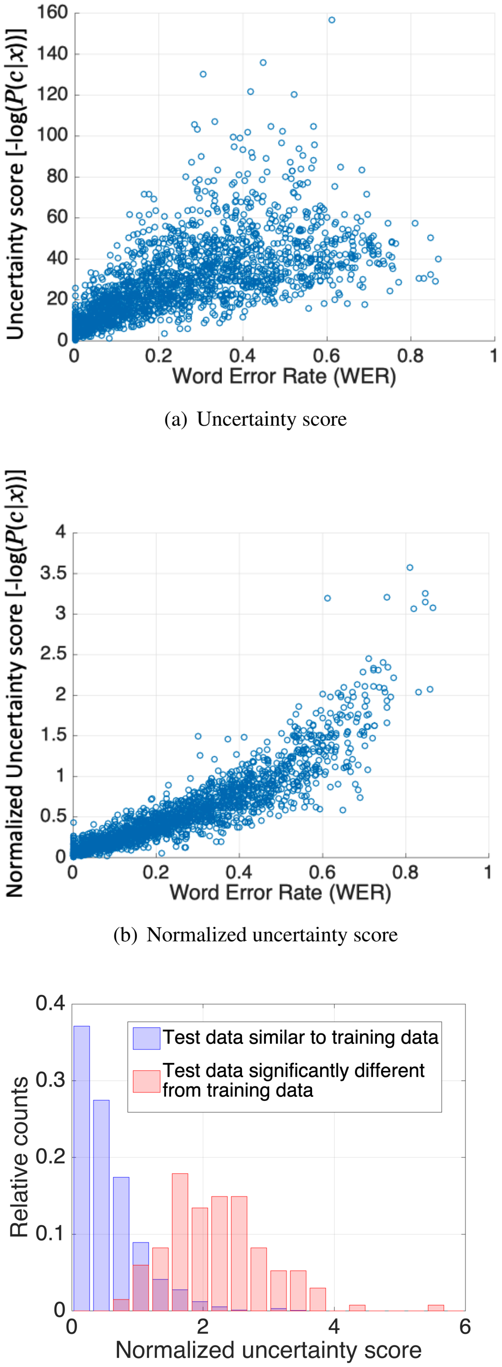
is valuable to know when the ASR models are expected to be accurate, and when they are not. To this end, we propose to use the *uncertainty score*, defined as the negative logarithm of the confidence score ($Q(c)$), as a surrogate for the word error rate. The confidence score of an utterance, and therefore its uncertainty score, depends on the character probabilities of the transcribed text, as determined by the acoustic model and language model of the ASR system. Figure 4(a) shows a scatter plot of the uncertainty score and word error rate, for each utterance in the test set. For the purpose of illustration, the uncertainty score was computed using only the acoustic model. The figure shows a clear correlation between the uncertainty score and the word error rate. This correlation increases further when one considers the normalized uncertainty score, defined as the uncertainty score divided by the number of characters in the transcription of the utterance. Figure 4(b) shows a scatter plot of the normalized uncertainty score and the word error rate, for the utterances in the test set. The normalized uncertainty score increases with the word error rate, and can be used to identify instances in which one would expect a high WER (i.e., low accuracy of the ASR model).

For applications in which the ASR model is used as part of a decision-support tool, one could flag, or even exclude, transcriptions that have a higher-than-average normalized uncertainty score. The uncertainty score can also be evaluated at the word-level, when the accuracy of the extracted operational information is a quantity of interest. The uncertainty score can also be used to identify off-nominal events during which we expect the performance of the speech recognition model to degrade (for example, because the speech-rate or the phraseology are significantly different from the nominal periods over which the model was trained).

Figure 4(c) shows the distribution of the normalized uncertainty score computed over two test datasets: (a) one that is similar to the dataset used for training the ASR model (shown in blue, and representative of conversations during nominal events); (b) a test dataset that is significantly different (in terms of accent and speech rate) from the training data set (shown in red, and representative of conversations during off-nominal events). We observe that the distribution of the normalized uncertainty score for the off-nominal data is significantly different (with a higher mean value, indicative of higher WERs) from the distribution obtained for the test dataset that is similar to the training data. Hypothesis testing techniques can be used to determine if an utterance (or a series of utterances) lies outside the nominal distribution in a statistical sense, and to flag those periods as off-nominal events.

Alternative language models

In this paper, we used a standard word N-gram language model for simplicity, and focused primarily on the acoustic model and the extraction of operational information. However, prior studies have shown that more sophisticated language models such as the class N-gram or RNN-based language models can improve the accuracy of ASR



(a) Uncertainty score

(b) Normalized uncertainty score

(c) Distribution of normalized uncertainty score

Figure 4. Uncertainty score of the transcription

systems (*6*, *27*). Another potential benefit of the class N-gram model is that one can efficiently incorporate out-of-vocabulary data, such as new airlines or navigational aids that are specific to a particular region, by updating the class definition file.

Semi-supervised learning

The availability of sufficient amounts of high-quality data is a key factor in improving the performance of ASR models. Our experiments have shown that the use of diverse datasets for model training results in more general models. However, there are not many openly-available corpora for ATC communications, and the manual transcription of ATC conversations requires significant effort. A different direction is to use semi-supervised learning, namely, to utilize widely-available but untranscribed ATC audio data (*38*) to improve the accuracy of the ASR model. The proposed approach would be as follows: (a) train a preliminary model with the existing ATC speech corpora; (b) use this preliminary model to transcribe utterances of ATC voice communications that does not have transcripts; (c) evaluate the accuracy of each of the utterances using the uncertainty score as a metric; and (d) re-train the ASR model by including the utterances that yielded a low uncertainty score in the training data. Although semi-supervised learning for ATC applications has been previously attempted (*25*), we believe that the use of the normalized uncertainty score to select data for retraining models is a promising direction for further investigation. Semi-supervised learning techniques (*39*) have been recently shown to perform well for regular conversational speech even with small amounts of labelled training data; these techniques could be extended to the ATC domain.

Acoustic model

In this paper, we have used Mozilla's implementation of DeepSpeech model that considered MFCC as the input feature. One could explore other alternative input features. For example, filter bank energy is an attractive alternate option given that there have been studies that have shown that they perform better than MFCCs for robust speech recognition (*40*). Further, recent studies have shown that Transformer-based architecture for the acoustic model yields better performance compared to conventional RNN, even for low resource languages (*41*). This makes Transformer-based model an attractive alternative for the ATC domain.

Another important aspect that one might have to consider is constraints on the model inference time depending on the application of interest. A larger neural network model (number of hidden layers and neurons in each layer) for the speech recognition system typically requires higher computational time for inference, resulting in higher RTF. On the other hand, a larger model might yield better accuracy. For a practitioner, it might be of interest to quantify this trade-off between RTF and accuracy, which is a potential direction for future research.

Conclusions

This paper investigated the use of an automatic speech recognition model for ATC voice communications data, and for the extraction of operational information from the same. The model followed an end-to-end speech recognition architecture, and was based on a recent machine learning model that involves recurrent neural network (Deep Speech). We were able to obtain a word error rate of 0.17 with our speech recognition model, on an independent test set of ATC communications data. The analysis revealed that including an N-gram language model in addition to the acoustic model improved the accuracy by 26%, and that transfer learning and parameter fine-tuning with a model pre-trained on conversational speech did not improve the accuracy. The results illustrated that as expected, the use of diverse data sources during training resulted in more generalizable models, i.e., models with better accuracy on test datasets from different sources. Recent ASR models (6, 24) have reported better performance in terms of WER compared to our proposed model. However, the intent of our research was to understand the impacts of the language model, parameter fine tuning, transfer learning, and training with additional data, and to consider the potential to extract operational information using natural language processing techniques.

To this end, we illustrated a call-sign extraction method using Named Entity Recognition, which yielded an F-1 score of 0.95 on the actual transcript, and an F-1 score of 0.69 on the transcript generated by the ASR model. The runway information was extracted using a rule-based grammar, resulting in an F-1 score of 1 (perfect transcription) on the actual transcript, and 0.95 on the transcript generated by the ASR model. We believe that the accuracy of call-sign extraction would further improve with more training data and better ASR accuracy. Further, we identified opportunities to improve the accuracy of ASR models for the ATC domain, including employing better language models, semi-supervised learning, incorporating better priors or context, and training the model with larger quantities of transcribed ATC speech data. These enhancements could potentially improve transcription accuracy, enabling practical applications ranging from real-time safety monitoring to speech assistants for air traffic controllers.

Acknowledgements

The authors would like to thank Azreen Zaman for processing some of the datasets (32–34) used to train the models, and Nadia Dimitrova for helpful discussions on potential language models for ATC communications. We also would like to thank Airbus for providing their ATC speech corpus (23), and Dr. Jim Glass for providing access to the NIST ATC Complete corpus (34). This work was supported by the ACRP Graduate Research Award Program, and has greatly benefited from discussions with Monica Alcabin, Robert Samis, Jonathan Rein and Larry Goldstein.

References

1. Kopald, H. D., A. Chanen, S. Chen, E. C. Smith, and R. M. Tarakan. Applying automatic speech recognition technology to air traffic management. In *2013 IEEE/AIAA 32nd Digital Avionics Systems Conference (DASC)*. IEEE, 2013, pp. 6C3–1.
2. Nguyen, V. N. and H. Holone. Possibilities, challenges and the state of the art of automatic speech recognition in air traffic control. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, Vol. 9, No. 8, 2015, pp. 1940–1949.
3. Amodei, D., S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*. 2016, pp. 173–182.
4. Subramanian, S. V., P. F. Kostiuk, and G. Katz. Custom IBM Watson speech-to-text model for anomaly detection using ATC-pilot voice communication. In *2018 Aviation Technology, Integration, and Operations Conference*. 2018, p. 3979.
5. Chen, S., H. Kopald, R. Chong, Y. Wei, and Z. Levonian. Read back error detection using automatic speech recognition. In *12th USA/Europe Air Traffic Management Research and Development Seminar (ATM2017), Seattle, WA, USA*. 2017.
6. Lin, Y., L. Deng, Z. Chen, X. Wu, J. Zhang, and B. Yang. A Real-Time ATC Safety Monitoring Framework Using a Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems*.
7. Helmke, H., J. Rataj, T. Mühlhausen, O. Ohneiser, H. Ehr, M. Kleinert, Y. Oualil, M. Schulder, and D. Klakow. Assistant-based speech recognition for ATM applications. In *11th USA/Europe Air Traffic Management Research and Development Seminar (ATM2015), Lisbon, Portugal*. 2015.
8. Šmídl, L., J. Švec, A. Pražák, and J. Trmal. Semi-supervised training of DNN-based acoustic model for ATC speech recognition. In *International Conference on Speech and Computer*. Springer, 2018, pp. 646–655.
9. Tarakan, R., K. Baldwin, and N. Rozen. An automated simulation pilot capability to support advanced air traffic controller training. In *The 26th Congress of ICAS and 8th AIAA ATIO*. 2008, p. 8897.
10. Lu, H.-L., V. H. Cheng, D. Ballinger, A. Fong, J. Nguyen, S. Jones, and S. E. Cowart. A Speech-Enabled Simulation Interface Agent for Airspace System Assessments. In *AIAA Modeling and Simulation Technologies Conference*. 2015, p. 0148.
11. Cordero, J. M., N. Rodríguez, J. M. de Pablo, and M. Dorado. Automated speech recognition in controller communications applied to workload measurement. *3rd SESAR Innovation Days, Stockholm, Sweden*.
12. Lowry, M., T. Pressburger, D. A. Dahl, and M. Dalal. Towards Autonomous Piloting: Communicating with Air Traffic Control. In *AIAA Scitech 2019 Forum*. 2019, p. 2207.
13. Rabiner, L. R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, Vol. 77, No. 2, 1989, pp. 257–286.

14. Gales, M. and S. Young. *The application of hidden Markov models in speech recognition*. Now Publishers Inc, 2008.
15. Morgan, N. and H. Bourlard. Continuous speech recognition. *IEEE signal processing magazine*, Vol. 12, No. 3, 1995, pp. 24–42.
16. Anon. The Bavieca ASR toolkit. http://www.bavieca.org/index.html, 2020. Retrieved on June 11th, 2020.
17. Lamere, P., P. Kwok, E. Gouvea, B. Raj, R. Singh, W. Walker, M. Warmuth, and P. Wolf. The CMU SPHINX-4 speech recognition system. In *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong*, Vol. 1. 2003, pp. 2–5.
18. Anon. Hidden Markov Model Toolkit. http://htk.eng.cam.ac.uk, 2020. Retrieved on June 11th, 2020.
19. Povey, D., A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, et al. The Kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011.
20. Hannun, A., C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.
21. Pratap, V., A. Hannun, Q. Xu, J. Cai, J. Kahn, G. Synnaeve, V. Liptchinsky, and R. Collobert. Wav2letter++: The Fastest Open-source Speech Recognition System. *CoRR*, Vol. abs/1812.07625. URL https://arxiv.org/abs/1812.07625.
22. Li, J., V. Lavrukhin, B. Ginsburg, R. Leary, O. Kuchaiev, J. M. Cohen, H. Nguyen, and R. T. Gadde. Jasper: An end-to-end convolutional neural acoustic model. *arXiv preprint arXiv:1904.03288*.
23. Delpech, E., M. Laignelet, C. Pimm, C. Raynal, M. Trzos, A. Arnold, and D. Pronto. A real-life, french-accented corpus of air traffic control communications. In *Language Resources and Evaluation Conference (LREC)*. 2018.
24. Pellegrini, T., J. Farinas, E. Delpech, and F. Lancelot. The airbus air traffic control speech recognition 2018 challenge: Towards ATC automatic transcription and call sign detection. *arXiv preprint arXiv:1810.12614*.
25. Srinivasamurthy, A., P. Motlicek, I. Himawan, G. Szaszak, Y. Oualil, and H. Helmke. *Semi-supervised learning with semantic knowledge extraction for improved speech recognition in air traffic control*. Tech. rep., 2017.
26. Oualil, Y., D. Klakow, G. Szaszák, A. Srinivasamurthy, H. Helmke, and P. Motlicek. A context-aware speech recognition and understanding system for air traffic control domain. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 404–408.
27. Nguyen, V. N. *Using linguistic knowledge for improving automatic speech recognition accuracy in air traffic control*. Master's thesis, 2016.
28. Wang, D., X. Wang, and S. Lv. An overview of end-to-end automatic speech recognition. *Symmetry*, Vol. 11, No. 8, 2019, p. 1018.
29. Mozilla Corporation. DeepSpeech Model. https://deepspeech.readthedocs.io/en/r0.9/, 2021. Retrieved on May 7th, 2021.
30. Kawakami, K. Supervised sequence labelling with recurrent neural networks. *Ph. D. dissertation, PhD thesis. Ph. D. thesis*.
31. Heafield, K. KenLM: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*. Association for Computational Linguistics, 2011, pp. 187–197.
32. Eurocontrol. https://www.eurocontrol.int/eec/public/standard_page/EEC_News_2008_1_ATCOSIM.html, 2019. Retrieved Sept 9, 2019.
33. Anon, University of West Bohemia, Department of Cybernetics. https://lindat.mff.cuni.cz/repository/xmlui/handle/11858/00-097C-0000-0001-CCA1-0?show=full, 2019. Retrieved Sept 9, 2019.
34. Godfrey, John. Air Traffic Control Complete LDC94S14A. Web Download. Philadelphia: Linguistic Data Consortium, 1994, 2020. Retrieved Sept 9, 2019.
35. Pan, S. J. and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, Vol. 22, No. 10, 2009, pp. 1345–1359.
36. Kunze, J., L. Kirsch, I. Kurenkov, A. Krug, J. Johannsmeier, and S. Stober. Transfer learning for speech recognition on a budget. *arXiv preprint arXiv:1706.00290*.
37. Spacy. https://spacy.io, 2020. Retrieved Jun 19, 2020.
38. https://www.liveatc.net, 2019. Retrieved Aug 21, 2019.
39. Baevski, A., H. Zhou, A. Mohamed, and M. Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*.
40. Paliwal, K. K. On the use of filter-bank energies as features for robust speech recognition. In *ISSPA'99. Proceedings of the Fifth International Symposium on Signal Processing and its Applications*, Vol. 2. IEEE, 1999, pp. 641–644.
41. Karita, S., N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang, et al. A comparative study on transformer vs RNN in speech applications. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 449–456.