Speech and Natural Language Processing Technologies for Pseudo-Pilot Simulator

Amrutha Prasad⋆,†,1,2, Juan Zuluaga-Gomez†,1,3, Petr Motlicek1,2, Saeed Sarfjoo1,
Iuliia Nigmatulina1,4, Karel Vesely2

1Idiap Research Institute, Martigny, Switzerland
2Brno University of Technology, Speech@FIT, IT4I CoE, Brno, Czech Republic
3Ecole Polytechnique Federale de Lausanne (EPFL), Lausanne, Switzerland
4Institute of Computational Linguistics, University of Zurich, Switzerland
⋆corresponding author: amrutha.prasad@idiap.ch
†equal contribution

Abstract—This paper describes a simple yet efficient repetition-based modular system for speeding up air-traffic controllers (ATCos) training. E.g., a human pilot is still required in EUROCONTROL's ESCAPE lite simulator `https://www.eurocontrol.int/simulator/escape` during ATCo training. However, this need can be substituted by an automatic system that could act as a pilot. In this paper, we aim to develop and integrate a pseudo-pilot agent into the ATCo training pipeline by merging diverse artificial intelligence (AI) powered modules. The system understands the voice communications issued by the ATCo, and, in turn, it generates a spoken prompt that follows the pilot's phraseology to the initial communication. Our system mainly relies on open-source AI tools and air traffic control (ATC) databases, thus, proving its simplicity and ease of replicability. The overall pipeline is composed of the following: (1) a submodule that receives and pre-processes the input stream of raw audio, (2) an automatic speech recognition (ASR) system that transforms audio into a sequence of words; (3) a high-level ATC-related entity parser, which extracts relevant information from the communication, i.e., callsigns and commands, and finally, (4) a speech synthesizer submodule that generates responses based on the high-level ATC entities previously extracted. Overall, we show that this system could pave the way toward developing a real proof-of-concept pseudo-pilot system. Hence, speeding up the training of ATCos while drastically reducing its overall cost.

Index Terms—Machine learning, air traffic controller training, air traffic management, BERT, automatic speech recognition, speech synthesis

I. INTRODUCTION

The communication between air-traffic controllers (ATCos) and pilots often requires the ATCos to issue commands for the safe travel of an aircraft. Although different means of communication such as CPDLC[1] messages, flight strips, or shortcodes are used, voice is the central part of these communications. Recent research has focused on improving the assistant-based speech recognition system (ABSR) for ATCos to reduce overall workload while increasing safety [1],

[2]. Another application in which speech recognition can be used in is the training of the ATCos. Training air-traffic controllers (ATCos) usually require a human pseudo-pilot. The pseudo-pilots respond or issues requests to the ATCo trainee to simulate a standard ATC communication scenario for an ATCo. The ATCo training is a human-intensive task, specialized workforce, and the overall cost is usually high. The pseudo-pilots are required to follow the commands issued by the ATCos. Also, they update the simulator so the ATCos can see if the pilots follow the orders similar to the actual situation.

An autonomous pilot agent could aid different parts of ATCo's training process to reduce the training costs due to specialized human involvement. For instance, previous research [3], [4] has investigated deep learning based framework for only repetition generator. In this work, we develop a modular, simple, and easy-to-deploy artificial intelligence (AI) based system for simulating a pilot, i.e., a pseudo-pilot. The main focus is to develop a pipeline from transcribing a controller's speech to generating the entities, which produces a response to simulate a pilot. Figure 1 describes the overall pipeline used in our system. The first module uses an automatic speech recognition (ASR) system to convert the voice communication issued by the ATCo trainee into text. The following module is a Named Entity Recognition (NER) system[2] that maps the text to different ATC-related categories such as callsigns, commands, and values. The last module is a repetition generator that returns the pilot response based on ATCo's voice message. This module consists of two submodules: (i) a rule-based system to convert the text into an appropriate pilot response, and (ii) a text-to-speech (TTS) system[3] that generates the corresponding response into a spoken form, i.e., speech play in the headphones of the ATCO trainee.

The proposed system handles the most frequent commands in ATC communications (see a list of commands and words in Figure 1). In summary, the proposed pseudo-pilot agent is well

---

[1]Controller Pilot Data Link Communications (CPDLC). CPDLC is a two-way data-link system by which controllers can transmit non-urgent strategic messages to an aircraft as an alternative to voice communications. The message is displayed on a flight deck visual display.

[2]We also refer to it as 'high-level entity parser' in several parts of the paper.
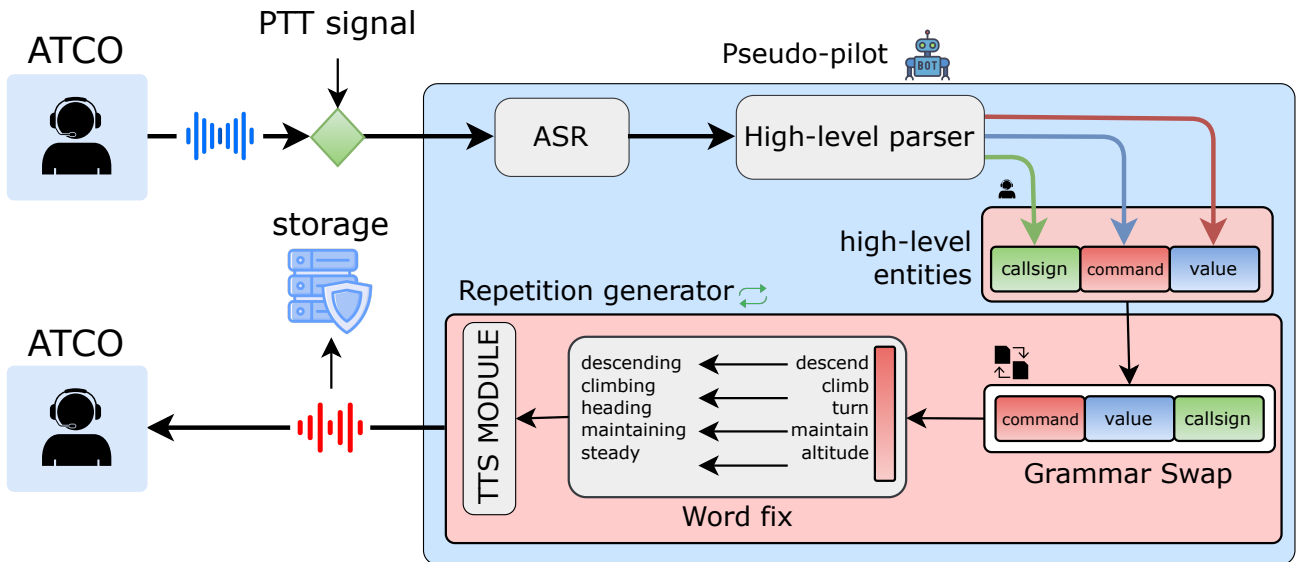[3]Also known as a speech synthesizer system.

Figure 1. **Overall pipeline for simulating a pilot for air traffic controller training.** The pipeline starts with an ATCo issuing a communication and its capture after the end of the PTT signal. Then, the speech-to-text and high-level entity parser modules transcribe and extract the ATC-related entities from the voice communication. The output is then accustomed to 'pilots' structured grammar. The speech synthesizer then uses the generated text to create a WAV file containing the spoken textual prompt. In the end, a pseudo-pilot response is produced that can provide pertinent feedback to ATCo trainees.

suited for the early stages of ATCo training, where potential controllers are required to build fundamental knowledge and skills. The proposed system can be adapted easily to specific conditions based on different airports where the spoken language is foreign. Finally, our system is versatile due to its simplicity and modularity nature. For example, we can plug in or out different modules depending on the need of the specific training setup.

The rest of the paper is organized as follows. Section II briefly describes previous research in the field of ASR, NER and TTS systems. This is followed by Section III that describes the data used in this work for training and evaluating different systems. Section IV describes the details of the modules used in this research and presents experimental results for ASR and NER systems. This is followed by conclusions in Section V.

II. RELATED WORK

A. Air-traffic Controller Training System

Air traffic control communications between ATCos and pilots are crucial for the safe travel of an aircraft. The prominent part of this communication comes from the ATCos. They issue commands from tens to hundreds of aircraft in a short period, and this number varies depending on the airspace condition and period of the year (e.g., ATC operations shows increased workload during summer for ATCos). Hence, ATCo trainees must learn how to handle stressful and complex airspace situations. They also undergo different training procedures, for instance, communication with pilots, and this often happens in a simulation environment with human pseudo-pilots.

As previously covered in [5], the ATCos have 3 stages of training: (i) initial, (ii) operational, and (iii) continuation

training. As the training of controllers is a crucial component in air-traffic management (ATM), efforts have been made to develop simulation devices for the same [6]–[8]. Research also focused on the post-evaluation of this simulation-based training [9], [10] and optimizing the training procedure [11] which still includes the involvement of human pseudo-pilot and the cost of training is still high. However, in the recent works of [3], [4], the authors develop a deep learning-based framework for the repetition generator to implement an autonomous pilot agent (APA). In [3], the authors develop APA with the main focus on repetition generator and text-to-speech system. The authors used a sequence-to-sequence mapping for the repetition generator and a transformer model [12] to generate speech.

B. Automatic Speech Recognition

Automatic Speech Recognition (ASR) also termed a speech-to-text system converts speech in a given language to text. A standard ASR system employs an acoustic model (AM) and a language model (LM) to achieve this task. The former, AM, is trained with a set of speech recordings with a corresponding text, also referred to as transcripts. The AM represents the relationship between a speech signal and phonemes, or other linguistic units, that make up the speech. The latter, LM, is trained on a large corpus of text data. The LM is usually represented by a probability distribution over sequences of words. The LM provides context to distinguish between words and phrases that sound similar. Using the knowledge of AM and LM, a decoding graph is usually built as a Weighted Finite State Transducer (WFST) [13]–[15]. The WFST graph generates text output given an observation sequence as shown in Figure 2.

Generally, recorded speech is represented as a sequence of acoustic feature vectors or *observations*, X; and the output word sequence, W. During recognition or *decoding*, the main goal is finding the most likely W given the input sequence X. To solve this task, statistical models are trained using a *corpus* $\mathcal{D}$ of labelled training *utterances*, (X^n, W^n). In general, if X is the sequence of acoustic features and W denotes a word sequence, the most likely word sequence is given by $\hat{W}$:

$$\hat{W} = arg \max_{W} P(W|X). \qquad (1)$$

The problem is further reformulated using Bayes Theorem:

$$P(W|X) = \frac{p(X|W)p(W)}{p(X)}, \qquad (2)$$

where, $P(X|W)$ stands for the likelihood of the feature sequence X, given the hypothesized word sequence, W. $P(W)$ is the probability of the word sequence (normally, computed from a pretrained LM). $P(X)$ is the a-priori probability of the feature sequence X, but it is ignored during the maximization operation due to its non-dependency to the AM and LM. Equation 3 is further simplified as $P(X)$ is a constant for any word sequence, as follows:

$$\hat{W} = arg \max_{W} p(X|W) \, p(W). \qquad (3)$$

Standard ASR systems rely on a lexicon, language, and acoustic model as stated above. Currently, there are two main ASR paradigms, where different strategies, architectures, and procedures are employed for blending all these modules in one "system".

1) Hybrid Based ASR: Automatic speech recognition with hybrid systems is based on Hidden Markov Models (HMM) and Deep Neural Networks (DNN) [16], [17]. DNNs are an effective module for the estimating the posterior probability of a given set of possible outputs (e.g., phone-state or tri-phone state probability estimator, in ASR systems). These posterior probabilities can be seen as pseudo-likelihoods or "scale likelihoods", which can be interfaced with HMM modules. HMMs provide a structure for mapping a temporal sequence of acoustic features, X, e.g., Mel-frequency cepstral coefficients (MFCCs) into a sequence of states [18], [19]. Hybrid systems remain one of the best approaches for building ASR engines. Currently, HMM-DNNs based ASR is the state-of-the-art systems for ASR in ATC domain [20]–[22].

Moreover, recent work in ASR has targeted different areas in ATC. For instance, a benchmark for ASR on ATC communications databases is established in [23]. Leveraging non-transcribed ATC audio data by semi-supervised learning has been covered in [20], [21], [24]. Previous work related to the large-scale automatic collection of ATC audio data from different airports worldwide is covered in [22], [25]. Additionally, innovative research targeted to improve callsign recognition by integrating surveillance data into the pipeline is covered by [26]–[28].
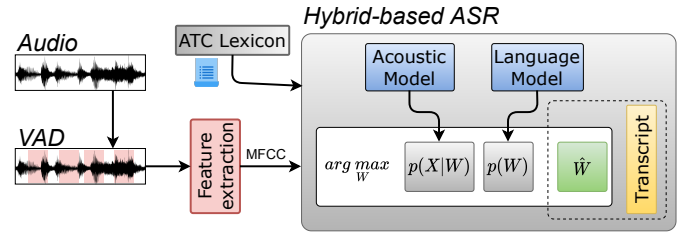

Figure 2. **Traditional automatic speech recognition system based on hidden Markov models and deep neural networks**. The system receives as input an ATCo voice communication, and it produces transcripts as output.

As depicted in Figure 2, the main components of a hybrid system are: a pronunciation lexicon, language model, and acoustic model. One key advantage of a hybrid system versus other ASR techniques is that the text data (e.g., words, dictionary) and pronunciation of new words are collected and added beforehand, hoping to match the target domain of the recognizer. Standard hybrid-based ASR approaches still rely on word-based lexicons, i.e., missing or out-of-vocabulary words from the lexicon cannot be hypothesized by the ASR decoder.

2) End-to-End ASR: End-to-end (E2E) speech recognition is another paradigm for performing ASR. E2E-ASR aims at directly transcribing speech to text without requiring alignments between acoustic frames (i.e., input features) and output characters/words, which is normally required in hybrid-based ASR (see Section II-B1). Unlike the hybrid approaches, the E2E model learns a direct mapping between acoustic frames and character units or words in one step towards the final objective of interest.

Recent work on encoder-decoder ASR can be categorized in two main approaches: Connectionist Temporal Classification (CTC) [29] and attention-based encoder-decoder systems [30]. CTC uses intermediate label representation, allowing repetitions of labels and occurrences of the so-called 'blank output' to label an output with 'no label'. Attention-based encoders-decoders directly learn a mapping from input acoustic frames to character sequences. At each output time step, the model emits a character unit conditioned on the inputs and the history of the produced outputs. Related work on self-supervised learning [31] for speech representation covers bidirectional models [32], [33] and autoregressive models [34], [35]. An innovative research on E2E-ASR for ATC domain is covered in [36] in which the authors fine-tuned a Wav2Vec 2.0 model [32] with public and private ATC databases, reaching on-par performances with hybrid-based ASR models.

C. High-Level Entity Parser

A high-level entity parser[4] system has the task to detect, classify and extract keywords from a given snippet of text or transcribed ATC communication. These keywords normally

---

[4]In a more technical domain, for instance, natural language processing, the task above-mentioned is called named-entity recognition (NER). For simplicity and because it is more aligned to ATC domain, we term it 'high-level entity parsing'.
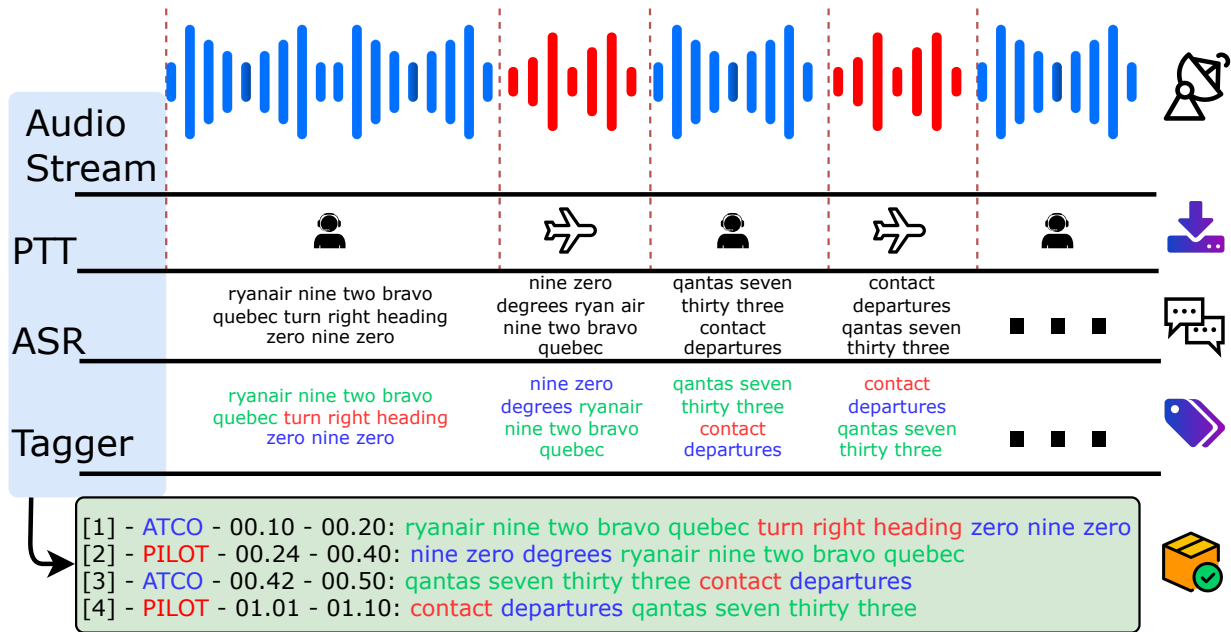
Figure 3. **Detailed outputs of the main ML-based submodules of our proposed pseudo-pilot system**. It includes pre-processing from the input audio stream, speaker role detection by push-to-talk (PTT) signal, transcripts generation and high-level ATC entities extraction with our speech-to-text and NER modules, respectively. All the data is later aggregated, packaged and sent to the repetition generator and TTS module. Note that this data can also be logged into a database for control and record.

fall into certain pre-defined categories such as parts of speech, location, organizations, or proper nouns like persons' names. Based on the domain, these entities can differ. In the field of ATC we have defined as 'key entities': callsigns, commands, values, and units. For example, the following transcribed communication (example taken from Figure 3):

Input: ryanair nine two bravo quebec turn right heading zero nine zero,

would be parsed to high-level entity format:

Output: <callsign> ryanair nine two bravo quebec </callsign> <command> turn right heading </command> <value> zero nine zero </value>

Early research [37] on high-level entity parsing or 'NER', used to obtain these tags from handcrafted dictionaries and ontologies. This, in turn, increased the overall complexity and was prone to human errors when escaling up to more entities or when adapting the system to a different domain. Collobert et al. [38] introduced machine learning-based methods for text processing in topics such as part-of-speech tagging, chunking, NER, and semantic role labeling. Further work on NER was carried by [39], [40]. In practice, a high-level entity parser system can be developed by fine-tuning a pre-trained LM on the downstream NER task. State-of-the-art NER systems are based on well-known pretrained LMs such as, BERT [41], RoBERTa [42], or DeBERTa [43]. In our experiments, we adopted one of the most well-known and simple to use LM, i.e., BERT.

D. Speech Synthesis

Speech synthesis also known as text-to-speech (TTS) is a technology involving several research fields such as linguistics, acoustics, speech signal processing, among others. TTS aims at transforming input text into a speech signal. There have been several approaches in the framework of TTS such as, formant-based parametric synthesis [44], waveform concatenation [45], or SPSS-based[5] models [46]. Akin to language modeling, recent advances in deep learning has also impacted TTS. For instance, the widely known Tacotron model [47] was proposed in 2017 and Tacotron2 [48], an updated version, in 2018. These models are end-to-end generative TTS systems capable to synthesize speech from characters (or words). More recently, FastSpeech2 [49] has gained a lot of popularity in the TTS research field due to its simplicity and because it works in a non-autoregressive manner. We refer the reader to the references cited above to get a more technical background behind these state-of-the-art end-to-end TTS systems.

III. DATASETS

This section briefly describes the public and private databases used for training and evaluating the ASR and high-level entity parser modules. Table I lists a summary of the databases employed in this research. It also lists the amount of train/test samples and their open-source status.

LDC-ATCC corpus: the Air Traffic Control Corpus[6] (ATCC) consists of recorded speech for use in ATC research

---

[5]SPSS stands for statistical parametric speech synthesis.
[6]LDC-ATCC: `https://catalog.ldc.upenn.edu/LDC94S14A`.

TABLE I. **Air traffic control communications-related databases used for developing our systems.** †total number of hours of audio after silence removal.

Database	Duration†		Open source	Ref
	Train	Test		
Private databases				
HAAWAII	43	4	✗	[36]
Internal Data	95	-	✗	[23]
Public databases				
ATCOSIM	8	-	✓	[50]
UWB-ATCC	10.4	-	✓	[51]
LDC-ATCC	23	2.6	✓	[52]
ATCO2-PL	100	4	✓	[25]

in the area of ASR and NLP. The audio data contains voice communication traffic between various ATCos and pilots. The audio files are sampled at 8 kHz, 16-bit linear, representing continuous monitoring without squelch or silence elimination. Each file has a single frequency over one to two hours of audio. The corpus contains gold annotations and metadata.[7] The corpus consists of approximately 70 h and after silence removal, the total duration of data is around 25 h of ATCo and pilot transmissions.

UWB-ATCC corpus: the UWB-ATCC[8] corpus is a free and public resource for research on ATC. It contains recordings of communication between ATCos and pilots. The speech is manually transcribed and labeled with the speaker information, i.e., pilot/controller. The total amount of speech after removing silences is 13 hrs. The audio data is mono-channel sampled at 8kHz and 16-bit PCM.

ATCO2 corpus: dataset built for the development and evaluation of ASR and NLP technologies for English ATC communications. The dataset consists of English coming from LKTB, LKPR, LZIB, LSGS, LSZH, LSZB and YSSY airports. There are two official partitions, namely, *ATCO2 test set 1h corpus* and the *ATCO2 test set corpus*. The first corpus contains 1.1 hr of open-sourced transcribed annotations, and it can be accessed for free in `https://www.atco2.org/data`. The latter contains around 3 hrs of extra annotated data, and the full corpus is available for purchase through ELDA in `http://catalog.elra.info/en-us/repository/browse/ELRA-S0484`. The recordings of both corpus are mono-channel sampled at 16kHz and 16-bit PCM.

HAAWAII corpus[9]: dataset based on an exploratory research project that aims to research and develop a reliable and adaptable solution to automatically transcribe voice commands issued by both ATCos and pilots. The controller

[7]Metadata covers voice activity segmentation details, speaker role information (who is talking), and callsigns in ICAO format.

[8]Corpus released by the University of West Bohemia: `https://lindat.mff.cuni.cz/repository/xmlui/handle/11858/00-097C-0000-0001-CCA1-0`.

[9]Highly Advanced Air Traffic Controller Workstation with Artificial Intelligence Integration: `https://www.haawaii.de`.
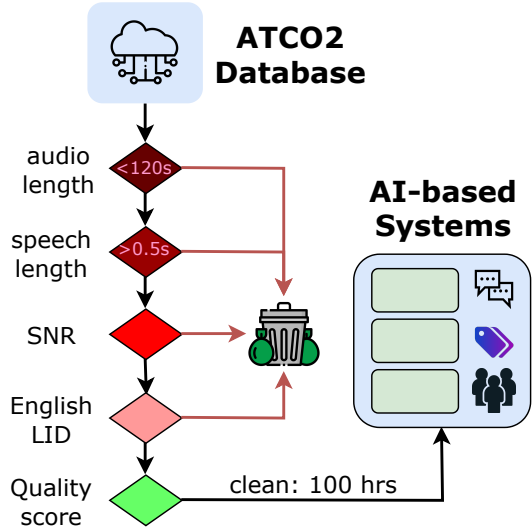

Figure 4. **Pipeline breakdown of data selection and filtering for training of the speech-to-text system**. SNR: signal-to-noise ratio; English LID: English language identification score. We apply these filters on top of the released version of *ATCO2 pseudo labeled set corpus* available for purchase through ELDA in `http://catalog.elra.info/en-us/repository/browse/ELRA-S0484`.

and pilot conversations are obtained from two Air Navigation Service Providers (ANSPs): (i) NATS for London Approach and (ii) ISAVIA for Icelandic en-route. The total amount of manually transcribed data available is around 47 h (partitioned into 43 h for train and 4 h for test). Similar to another corpus, the audio files are sampled at 8 kHz and 16-bit PCM.

IV. Air-traffic Controller Training System

In this section, we describe the core modules and present the obtained results of the proposed pseudo-pilot system. The first part covers pre-processing, automatic speech recognition, and high-level ATC-entity parsing, as described in Figure 3. The final output of these modules then creates the spoken sentence using the speech synthesizer to simulate a pilot (second part). Figure 1 gives a broad overview of the system.

A. Pre-processing

As mentioned in Section III, the sampling rate of all databases are not the same. Thus, the audio data is first up/down sampled to 16kHz. For ATCOs' data, the following steps are applied to select the best audio data from the open-source *ATCO2 PL set corpus*[10], which is described in Figure 4. Firstly, the available data is filtered to remove very long (>120 s) and short recordings (<0.5 s). Next, we remove segments with SNR below 0 dB in order to have reasonably good audio quality. As described previously, the ATCO2 corpus consists of data from different airports, and thus, there are some non-English recordings. However, our ASR and NLP systems are trained with only English data. Thus, in the next step, we apply English language detection

[10]The *ATCO2 PL set corpus* is composed of generated ASR transcripts, i.e., it is not manually annotated or corroborated by a human.

TABLE II. **COMPARISON OF WORD ERROR RATES (WER) IN PERCENTAGES FOR AM AND LM TRAINED WITH DIFFERENT ATC SUBSETS DESCRIBED IN SECTION III.** (I) MODEL 1- ALL DATA EXCEPT ATCO2-PL (190 H), (II) MODEL 2- TRAINED WITH PUBLIC DATASETS (LDC-ATCC + UWB-ATCC = 33 H), AND (III) MODEL 3- ONLY ATCO2-PL (100 H). EACH MODEL IS EVALUATED SEPARATELY ON HAAWAII, LDC-ATCC AND *ATCO2 test set corpus*. WERS IN BOLD DENOTES THE TOP PERFORMANCE PER TEST SET.

Model	Test sets WER (%)		
	HAAWAII	LDC	ATCO2
Model 1	**10.3**	**13.5**	36.6
Model 2	39.8	14.3	44.8
Model 3	38.1	58.3	**29.6**

and remove the samples that have less than 0.5 points on English language score (see [53]).[11] Finally, we select ∼100 h subset for training our system.

B. Automatic Speech Recognition

In this work, we present results for hybrid-based ASR systems trained with the corpora listed in Table I.

Experimental details: in our experiments, conventional biphone Convolutional Neural Network (CNN) [54] + TDNN-F [55] based acoustic models trained with Kaldi [56] toolkit (i.e., nnet3 model architecture) are used. AMs are trained with the LF-MMI training framework, considered to produce state-of-the-art performance for hybrid ASR. In all the experiments, 3-fold speed perturbation with MFCCs and i-vectors features are used. Language model (LM) is trained as a statistical 3-gram model using the manual transcripts. The results are presented for the AM and LM trained with the following 3 scenarios: (i) all data except the ATCO2 dataset, (ii) public dataset (ATCOSIM + LDC-ATCC + UWB-ATCC), and (iii) 100 h of ATCO2-PL data. We tag these models in Table II as `Model-1`, `Model-2` and `Model-3`, respectively.

Results: to show the performance of the ASR on various subsets of data described in Section III, we report results of the ASR on 3 test sets, a private data set and 2 public datasets: (i) HAAWAII, (ii) LDC, and (iii) ATCO2 test set corpus. Each test set is a combination of both ATCo and pilot speech. As shown in Table II, the best performance for each test set is reached when the ASR is trained with the respective data. For reproducible experiments, ASR trained with open-source data can be used. An error-resilient ASR is critical in the ATM domain. In [57], the authors show that achieving a lower WER increases the command recognition performance. The results in Table II also show that ATC communications in different environments result in various performances. Thus, depending on the condition of the environment, a suitable model can be used. E.g., evaluating the ASR on the ATCo subset of the HAAWAII test set results in a WER of ∼4% which implies that 4 out of 100 words of an ATCo are wrong, which is acceptable in practical applications.

---

[11]This score ranges from 0 to 1. The higher the score, the more probability that the ATC communication is in English. The authors of ATCO2-PL database supply this information as metadata.
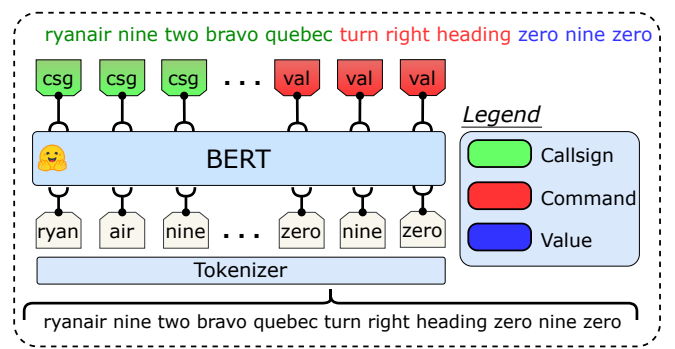

Figure 5. **High-level ATC-entity parser system**. This submodule is based on a pretrained BERT language model fetched from HuggingFace platform [59]. We fine-tune BERT in the named entity recognition task for identification of key entities in transcribed ATC communications.

C. High-level ATC-Entity Parser

Air traffic control communications follow a structured grammar and set of rules. Thus, it is logical to expect that some words and phases carry special meaning in the communication. Normally, these words uttered by ATCos are translated to some actions by pilots. We categorize three classes of entities following the *ATCO2 test set corpus* [22], [25], [58], i.e., callsigns, commands and values. We developed a baseline system for recognition of these entities based on NER, as depicted in Figure 5. We name this model: *High-level ATC-Entity Parser*. An early implementation of this system was covered in [28]. However, the authors only focus on private databases.

Datasets: we use the ATCO2 corpus for experiments. We split the initial subset into an 80/20 ratio to train and test our system. We did not use the other datasets from Table I because none of them contain gold annotations (i.e., annotations at word level) of high-level ATC-related entities.

Experimental details: a NER system is trained that parses text[12] into high-level entities relevant to ATC communications. The NER module is depicted in Figure 5. First, a BERT[13] [41] model is downloaded from HuggingFace [59], [60] which is then fine-tuned on the NER task with 3k sentences (∼3 hours of speech) using the *ATCO2 test set corpus*, where each word has a tag. The final layer of the BERT model is replaced by a linear layer with a dimension of 8 (following the classes structure from Section 3.3 of [61], two outputs for each class). As only 3k sentences are used, a 5-fold cross-validation is conducted. Further details about experimentation are covered in [28]. We redirected the reader to the public and open-source GitHub repository of the ATCO2 corpus (`https://github.com/idiap/atco2-corpus`). Here, the authors released Python scripts to replicate part of the results presented in this work.

---

[12]In our case, it parses transcripts generated by our ASR system or ground truth annotations as a proof-of-concept.
[13]The pre-trained version of `BERT-base-uncased` with 110 million parameters is used. URL: `https://huggingface.co/bert-base-uncased`.

TABLE III. MEAN AND STANDARD DEVIATION (STD) OF DIFFERENT PERFORMANCE METRICS FOR EACH ENTITY FROM OUR NER SYSTEM. PRECISION (INDICATES HOW ACCURATE THE MODEL IS AT WHAT IT PREDICTED), RECALL (HOW ACCURATE THE MODEL IS AT IDENTIFYING THE CORRECT CLASS), AND F1-SCORE METRICS ARE USED TO EVALUATE OUR HIGH-LEVEL ATC-ENTITY PARSER SYSTEM TRAINED AND EVALUATED ON ATCO2 TEST SET. A 5-FOLD CROSS-VALIDATION SCHEME IS CARRIED OUT, AS THE TRAIN AND TEST SPLIT ARE SMALL. RESULTS ARE PRESENTED FOR EACH CLASS, I.E., CALLSIGN, COMMAND, AND VALUE. NOTE THAT THE RESULTS ARE OBTAINED ON TOP OF GOLD ANNOTATIONS.

Class	Nb. Tokens	Metrics: Mean (Std)		
	Mean (Std)	Precision	Recall	F1-score
Callsign	2274 (45)	0.97 (0.004)	0.98 (0.004)	0.97 (0.005)
Command	1030 (7)	0.80 (0.016)	0.83 (0.023)	0.82 (0.020)
Values	2175 (49)	0.86 (0.008)	0.88 (0.015)	0.87 (0.006)

Results: Table III shows the results on ATCO2 test set. For each class, the results are presented using the precision, recall and F1-score metrics. The BERT-based model yielded an average of 0.97/0.82/0.87 F1-score for callsign, commands and values, respectively. We observe that the command class is the most challenging among all the classes, as they carry extra complexity compared to values and callsigns. For instance, a value is mainly composed of cardinal numbers (e.g., one, one hundred, one thousand) and some additional words, e.g., flight level. While a callsign is composed of an airline designator along with numbers and radiotelephony alphabet [62]. Overall, the recognition rates are above 80 points in all metrics (see Table III). However, in practical terms, further experimentation and validation needs to be undertaken with in real-life scenarios, in order to determine the minimum required performance to make the system viable. Similarly, there is still space for improvement, for instance, by adding real-time surveillance data into the system. An example is covered in [28]. Also, we should test the performance of the ATC-entity parser on top of ASR output rather than ground truths.

D. Repetition Generator

The repetition generator (RG) is the core of the pseudo-pilot agent. It receives the output of the NER system, which contains the callsign, commands, and values uttered by the ATCo, and it produces a spoken response. The response is a WAV file that is played back over the ATCos' trainee headphones. It can also be stored along with the metadata for future control and assessment. In essence, this response matches the grammar of what a typical pseudo-pilot (or pilot) would reply based on the initial commands issued by the ATCo. In addition, the RG system comprises three submodules: a grammar converter, a word fixer, and a text-to-speech module (also known as a speech synthesizer). An overview of the RG system is in the red box of Figure 1.

Grammar converter module: is built based on the fact that in ATC communications, pilots typically mention the callsign at the end, while ATCos do at the beginning. This module swaps the order of the entities detected by our NER system (see Section IV-C) to match the pilots' 'grammar' structure.

Word fixer module: it modifies the commands to match what a pseudo-pilot should reply.[14] We perform this by applying some mapping rules, e.g., descend → descending or turn → heading. These rules ensure that the reply generated by the RG is as close as possible to standard ICAO phraseology. Our current word fixer module contains a list of 15 commands. Adding additional mapping rules to a `rules.txt` file can easily update this module, allowing the system to work in different environments, e.g., ground tower or control approach.

Text-to-Speech module: finally, when the final textual prompt is assembled, an out-of-the-box TTS system converts the generated prompt into spoken format.

Experimental details: a recently proposed non-autoregressive speech synthesizer, FastSpeech2 model [49] is used in our experiments. A brief description of the details and implementation is in Section IV-D. We download and use out-of-the-box a well-known pre-trained TTS model. Specifically, we use FastSpeech2 model from HuggingFace hub [59], which is available at the following link: `https://huggingface.co/facebook/fastspeech2-en-ljspeech`. We use this model out-of-the-box by simply performing inference with the sentence produced by the repetition generator, i.e., the prompt of the pseudo-pilot. Additional models, such as Tacotron [47] or Tacotron2 [48],[15] can be also fine-tuned and deployed for the ATC data.

System analysis: during the experiments, we found out that the model can handle complicated word sequences[16], which are common in ATC. However, we did not perform any qualitative analysis of the produced voice/speech by the TTS. We leave this field as future line of work. Also, one can anticipate that akin to the NER system, the TTS module can be fine-tuned on the specific field, i.e., ATC. We did not explore this area, as the main idea of the paper was to implement a simple, yet efficient pseudo-pilot system with already available open-source models. Adapting the TTS system with ATC audio is also a future line of work.

V. CONCLUSIONS AND FUTURE WORK

In this work, we investigated an approach to generate a pseudo-pilot agent using speech and natural language processing techniques that are developed and implemented efficiently. The main modules comprise: transcribing ATCo communication with an ASR, parsing the ATC-related entities from the transcript with a NER module, and rendering a pilot-alike response using a simple repetition generator module. The ASR experimental results are presented for three different training scenarios: (i) using all available ATC data, (ii) using

---

[14] Based on the type of communication requirements, a similar approach can be deployed to update the response. e.g., 'create' a desirable read-back error. This can be useful in training ATCos to spot these errors. One example: `turn right` → `turn left`.

[15] Tacotron2 model is public and free to access in the HuggingFace hub in the following link: `https://huggingface.co/speechbrain/tts-tacotron2-ljspeech`.

[16] For example, pilot's read backs that include more than one command and values.

only open-source data, and (iii) using open-source but noisy data. These ASR models are then evaluated on three different test sets. The results show that each test set performs better when the data is seen during training, and it indicates that ASR is essential as the correct transcript is required to generate a pilot response. The high-level entity parser tags the sequence of words to callsigns, commands, and values are trained and evaluated on the ATCO2 test set and has F1-scores above 0.80 for all the classes. And, as the primary goal in this work is to generate a simple and efficient pseudo-pilot, we deploy a rule-based dialogue system along with a generic state-of-the-art TTS system to generate a pseudo-pilot agent.

We investigated hybrid-based systems for ASR, which can further be improved by incorporating (i) surveillance data as an additional modality and (ii) end-to-end training techniques. As mentioned earlier, the repetition generator uses a simple grammar converter and a pre-trained TTS system. As part of our future work, we consider investigating the grammar converter to (i) include greetings and complex responses and (ii) based on the surveillance data, to initiate a request from a pilot and TTS system to fine-tune it to the ATC domain. We also consider developing techniques to have a quantitative metric for evaluating the system.

REFERENCES

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

[2] O. Ohneiser, S. S. Sarfjoo, H. Helmke, S. Shetty *et al.*, "Robust command recognition for lithuanian air traffic control tower utterances," in *Interspeech 2021, Brno, Czechia, 30 August - 3 September 2021*. ISCA, 2021, pp. 3291–3295. [Online]. Available: https://doi.org/10.21437/Interspeech.2021-935

[3] Y. Lin, Y. Wu, D. Guo, P. Zhang *et al.*, "A deep learning framework of autonomous pilot agent for air traffic controller training," *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 5, pp. 442–450, 2021.

[4] J. Zhang, P. Zhang, D. Guo, Y. Zhou *et al.*, "Automatic repetition instruction generation for air traffic control training using multi-task learning with an improved copy network," *Knowledge-Based Systems*, vol. 241, p. 108232, 2022.

[5] I. D. Doc, "10056 manual on air traffic controller competency-based training and assessment first edition—2016."

[6] M. Pavlinović, B. Juričić, and B. Antulov-Fantulin, "Air traffic controllers' practical part of basic training on computer based simulation device," in *International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2017, pp. 920–925.

[7] B. Juričić, I. Varešak, and D. Božić, "The role of the simulation devices in air traffic controller training," in *International Symposium on Electronics in Traffic, ISEP 2011 Proceedings*, 2011.

[8] B. Chhaya, S. Jafer, W. B. Coyne, N. C. Thigpen *et al.*, "Enhancing scenario-centric air traffic control training," in *2018 AIAA modeling and simulation technologies conference*, 2018, p. 1399.

[9] A. W. Eide, S. S. Ødegård, and A. Karahasanović, "A post-simulation assessment tool for training of air traffic controllers," in *International Conference on Human Interface and the Management of Information*. Springer, 2014, pp. 34–43.

[10] H. Némethová, J. Bálint, and J. Vagner, "The education and training methodology of the air traffic controllers in training," in *International Conference on Emerging eLearning Technologies and Applications (ICETA)*. IEEE, 2019, pp. 556–563.

[11] J. A. Updegrove and S. Jafer, "Optimization of air traffic control training at the federal aviation administration academy," *Aerospace*, vol. 4, no. 4, p. 50, 2017.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems, December 4-9, 2017, Long Beach, CA, USA*, 2017, pp. 5998–6008.

[13] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech & Language*, vol. 16, no. 1, pp. 69–88, 2002.

[14] M. Mohri *et al.*, "Speech recognition with weighted finite-state transducers," in *Springer Handbook of Speech Processing*, 2008, pp. 559–584.

[15] M. Riley, C. Allauzen, and M. Jansche, "OpenFst: An open-source, weighted finite-state transducer library and its applications to speech and language," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Tutorial Abstracts*. Boulder, Colorado: Association for Computational Linguistics, 2009, pp. 9–10. [Online]. Available: https://aclanthology.org/N09-4005

[16] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on audio, speech, and language processing*, vol. 20, no. 1, pp. 30–42, 2011.

[17] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks." in *Interspeech*, vol. 2013, 2013, pp. 2345–2349.

[18] N. Morgan, H. Bourlard, S. Renals, M. Cohen *et al.*, "Hybrid neural network/hidden markov model systems for continuous speech recognition," in *Advances in Pattern Recognition Systems Using Neural Network Technologies*. World Scientific, 1993, pp. 255–272.

[19] H. A. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*. Springer Science & Business Media, 1993, vol. 247.

[20] A. Srinivasamurthy, P. Motlícek, I. Himawan, G. Szaszák *et al.*, "Semi-supervised learning with semantic knowledge extraction for improved speech recognition in air traffic control," in *Interspeech 2017, Stockholm, Sweden, August 20-24, 2017*. ISCA, 2017, pp. 2406–2410. [Online]. Available: http://www.isca-speech.org/archive/Interspeech_2017/abstracts/1446.html

[21] M. Kleinert, H. Helmke, G. Siol, H. Ehr *et al.*, "Semi-supervised adaptation of assistant based speech recognition models for different approach areas," in *37th Digital Avionics Systems Conference (DASC)*. IEEE, 2018, pp. 1–10.

[22] J. Zuluaga-Gomez, K. Veselý, A. Blatt, P. Motlicek *et al.*, "Automatic call sign detection: Matching air surveillance data with air traffic spoken communications," in *Multidisciplinary Digital Publishing Institute Proceedings*, vol. 59, no. 1, 2020, p. 14.

[23] J. Zuluaga-Gomez, P. Motlicek, Q. Zhan, K. Veselý *et al.*, "Automatic Speech Recognition Benchmark for Air-Traffic Communications," in *Interspeech*, 2020, pp. 2297–2301.

[24] J. Zuluaga-Gomez, I. Nigmatulina, A. Prasad, P. Motlicek *et al.*, "Contextual Semi-Supervised Learning: An Approach to Leverage Air-Surveillance and Untranscribed ATC Data in ASR Systems," in *Interspeech*, 2021, pp. 3296–3300.

[25] M. Kocour, K. Veselý, I. Szöke, S. Kesiraju *et al.*, "Automatic processing pipeline for collecting and annotating air-traffic voice communication data," *Engineering Proceedings*, vol. 13, no. 1, p. 8, 2021.

[26] M. Kocour, K. Veselý, A. Blatt, J. Zuluaga-Gomez *et al.*, "Boosting of contextual information in ASR for air-traffic call-sign recognition," in *Interspeech 2021, Brno, Czechia, 30 August - 3 September 2021*. ISCA, 2021, pp. 3301–3305. [Online]. Available: https://doi.org/10.21437/Interspeech.2021-1619

[27] I. Nigmatulina, R. Braun, J. Zuluaga-Gomez, and P. Motlicek, "Improving callsign recognition with air-surveillance data in air-traffic communication," *ArXiv preprint*, vol. abs/2108.12156, 2021. [Online]. Available: https://arxiv.org/abs/2108.12156

[28] I. Nigmatulina, J. Zuluaga-Gomez, A. Prasad, S. S. Sarfjoo *et al.*, "A two-step approach to leverage contextual data: speech recognition in air-traffic communications," in *ICASSP*, 2022.

[29] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, ser. JMLR Workshop and Conference Proceedings, vol. 32. JMLR.org, 2014, pp. 1764–1772. [Online]. Available: http://proceedings.mlr.press/v32/graves14.html

[30] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho *et al.*, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems, December 7-12, 2015, Montreal, Quebec, Canada*, 2015, pp. 577–585. [Online]. Available: https://proceedings.neurips.cc/paper/2015/hash/1068c6e4c8051cfd4e9ea8072e3189e2-Abstract.html

[31] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Interspeech 2019, Graz, Austria, 15-19 September 2019*. ISCA, 2019, pp. 3465–3469. [Online]. Available: https://doi.org/10.21437/Interspeech.2019-1873

[32] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems, December 6-12, 2020, virtual*, 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/92d1e1eb1cd6f9fba3227870bb6d7f07-Abstract.html

[33] S. Chen, C. Wang, Z. Chen *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *ArXiv preprint*, vol. abs/2110.13900, 2021. [Online]. Available: https://arxiv.org/abs/2110.13900

[34] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *ArXiv preprint*, vol. abs/1807.03748, 2018. [Online]. Available: https://arxiv.org/abs/1807.03748

[35] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=rylwJxrYDS

[36] J. Zuluaga-Gomez, A. Prasad, I. Nigmatulina, S. Sarfjoo *et al.*, "How does pre-trained wav2vec2.0 perform on domain shifted asr? an extensive benchmark on air traffic control communications," *IEEE Spoken Language Technology Workshop (SLT), Doha, Qatar*, 2023.

[37] R. Grishman and B. Sundheim, "Message Understanding Conference-6: A brief history," in *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996. [Online]. Available: https://aclanthology.org/C96-1079

[38] R. Collobert, J. Weston, L. Bottou, M. Karlen *et al.*, "Natural language processing (almost) from scratch," *Journal of machine learning research*, vol. 12, pp. 2493–2537, 2011.

[39] J. Piskorski, L. Pivovarova, J. Šnajder, J. Steinberger *et al.*, "The first cross-lingual challenge on recognition, normalization, and matching of named entities in Slavic languages," in *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing*. Valencia, Spain: Association for Computational Linguistics, 2017, pp. 76–85. [Online]. Available: https://aclanthology.org/W17-1412

[40] V. Yadav and S. Bethard, "A survey on recent advances in named entity recognition from deep learning models," in *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, 2018, pp. 2145–2158. [Online]. Available: https://aclanthology.org/C18-1182

[41] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[42] Y. Liu, M. Ott, N. Goyal, J. Du *et al.*, "Roberta: A robustly optimized bert pretraining approach," *ArXiv preprint*, vol. abs/1907.11692, 2019. [Online]. Available: https://arxiv.org/abs/1907.11692

[43] P. He, X. Liu, J. Gao, and W. Chen, "Deberta: decoding-enhanced bert with disentangled attention," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. [Online]. Available: https://openreview.net/forum?id=XPZIaotutsD

[44] D. H. Klatt, "Review of text-to-speech conversion for english," *The Journal of the Acoustical Society of America*, vol. 82, no. 3, pp. 737–793, 1987.

[45] I. R. Murray, J. L. Arnott, and E. A. Rohwer, "Emotional stress in synthetic speech: Progress and future directions," *Speech Communication*, vol. 20, no. 1-2, pp. 85–91, 1996.

[46] K. Tokuda, Y. Nankaku, T. Toda, H. Zen *et al.*, "Speech synthesis based on hidden markov models," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1234–1252, 2013.

[47] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu *et al.*, "Tacotron: Towards end-to-end speech synthesis," in *Interspeech 2017, Stockholm, Sweden, August 20-24, 2017*. ISCA, 2017, pp. 4006–4010. [Online]. Available: http://www.isca-speech.org/archive/Interspeech_2017/abstracts/1452.html

[48] J. Shen, R. Pang, R. J. Weiss, M. Schuster *et al.*, "Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*. IEEE, 2018, pp. 4779–4783. [Online]. Available: https://doi.org/10.1109/ICASSP.2018.8461368

[49] Y. Ren, C. Hu, X. Tan, T. Qin *et al.*, "Fastspeech 2: Fast and high-quality end-to-end text to speech," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. [Online]. Available: https://openreview.net/forum?id=piLPYqxtWuA

[50] K. Hofbauer, S. Petrik, and H. Hering, "The ATCOSIM corpus of non-prompted clean air traffic control speech," in *LREC*. Marrakech, Morocco: European Language Resources Association (ELRA), 2008. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2008/pdf/545_paper.pdf

[51] L. Šmídl, J. Švec, D. Tihelka, J. Matoušek *et al.*, "Air traffic control communication (ATCC) speech corpora and their use for ASR and TTS development," *Language Resources and Evaluation*, vol. 53, no. 3, pp. 449–464, 2019.

[52] J. Godfrey, "The Air Traffic Control Corpus (ATC0) - LDC94S14A," 1994. [Online]. Available: https://catalog.ldc.upenn.edu/LDC94S14A

[53] J. Zuluaga-Gomez, K. Veselý, I. Szöke, P. Motlicek *et al.*, "ATCO2 corpus: A Large-Scale Dataset for Research on Automatic Speech Recognition and Natural Language Understanding of Air Traffic Control Communications," *arXiv preprint arXiv:2211.04054*, 2022.

[54] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[55] D. Povey, G. Cheng, Y. Wang, K. Li *et al.*, "Semi-orthogonal low-rank matrix factorization for deep neural networks," in *Interspeech 2018, Hyderabad, India, 2-6 September 2018*. ISCA, 2018, pp. 3743–3747. [Online]. Available: https://doi.org/10.21437/Interspeech.2018-1417

[56] D. Povey *et al.*, "The kaldi speech recognition toolkit," in *IEEE workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011.

[57] H. Helmke, S. Shetty, M. Kleinert, O. Ohneiser *et al.*, "Measuring speech recognition and understanding performance in air traffic control domain beyond word error rates."

[58] M. Rigault, C. Cevenini, K. Choukri, M. Kocour *et al.*, "Legal and ethical challenges in recording air traffic control speech," in *Proceedings of the Workshop on Ethical and Legal Issues in Human Language Technologies and Multilingual De-Identification of Sensitive Data In Language Resources within the 13th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, 2022, pp. 79–83. [Online]. Available: https://aclanthology.org/2022.legal-1.14

[59] T. Wolf, L. Debut, V. Sanh, J. Chaumond *et al.*, "Transformers: State-of-the-art natural language processing," in *EMNLP*. Online: Association for Computational Linguistics, 2020, pp. 38–45. [Online]. Available: https://aclanthology.org/2020.emnlp-demos.6

[60] Q. Lhoest, A. Villanova del Moral, Y. Jernite, A. Thakur *et al.*, "Datasets: A community library for natural language processing," in *EMNLP*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 175–184. [Online]. Available: https://aclanthology.org/2021.emnlp-demo.21

[61] J. Zuluaga-Gomez, S. S. Sarfjoo, A. Prasad, I. Nigmatulina *et al.*, "Bertraffic: Bert-based joint speaker role and speaker change detection for air traffic control communications," *IEEE Spoken Language Technology Workshop (SLT), Doha, Qatar*, 2023.

[62] International Civil Aviation Organization, "ICAO Phraseology Reference Guide," 2020. [Online]. Available: https://www.skybrary.aero/bookshelf/books/115.pdf