aerospace

MDPI

Article

A Virtual Simulation-Pilot Agent for Training of Air Traffic Controllers

Juan Zuluaga-Gomez [1,2,*], Amrutha Prasad [1,3], Iuliia Nigmatulina [1,4], Petr Motlicek [1,4] and Matthias Kleinert [5]

[1] Speech & Audio Processing Group, Idiap Research Institute, 1920 Martigny, Switzerland; aprasad@idiap.ch (A.P.); iuliia.nigmatulina@idiap.ch (I.N.)
[2] LIDIAP, Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland
[3] Faculty of Information Technology, Brno University of Technology, 60190 Brno, Czech Republic
[4] Institute of Computational Linguistics, University of Zurich, 8006 Zurich, Switzerland
[5] Institute of Flight Guidance, German Aerospace Center (DLR), 38108 Braunschweig, Germany; matthias.kleinert@dlr.de
* Correspondence: juan-pablo.zuluaga@idiap.ch

Abstract: In this paper we propose a novel virtual simulation-pilot engine for speeding up air traffic controller (ATCo) training by integrating different state-of-the-art artificial intelligence (AI)-based tools. The virtual simulation-pilot engine receives spoken communications from ATCo trainees, and it performs automatic speech recognition and understanding. Thus, it goes beyond only transcribing the communication and can also understand its meaning. The output is subsequently sent to a response generator system, which resembles the spoken read-back that pilots give to the ATCo trainees. The overall pipeline is composed of the following submodules: (i) an automatic speech recognition (ASR) system that transforms audio into a sequence of words; (ii) a high-level air traffic control (ATC)-related entity parser that understands the transcribed voice communication; and (iii) a text-to-speech submodule that generates a spoken utterance that resembles a pilot based on the situation of the dialogue. Our system employs state-of-the-art AI-based tools such as Wav2Vec 2.0, Conformer, BERT and Tacotron models. To the best of our knowledge, this is the first work fully based on open-source ATC resources and AI tools. In addition, we develop a robust and modular system with optional submodules that can enhance the system's performance by incorporating real-time surveillance data, metadata related to exercises (such as sectors or runways), or even a deliberate read-back error to train ATCo trainees to identify them. Our ASR system can reach as low as 5.5% and 15.9% absolute word error rates (WER) on high- and low-quality ATC audio. We also demonstrate that adding surveillance data into the ASR can yield a callsign detection accuracy of more than 96%.

Keywords: air traffic controller training; simulation-pilot agent; BERT; automatic speech recognition and understanding; speech synthesis.

Citation: Zuluaga-Gomez, J.; Prasad, A.; Nigmatulina, I.; Motlicek, P.; Kleinert, M. A Virtual Simulation-Pilot Agent for Training of Air Traffic Controllers. *Aerospace* **2023**, *10*, 490. https://doi.org/10.3390/aerospace10050490

Academic Editor: Judith Rosenow

Received: 13 April 2023
Revised: 5 May 2023
Accepted: 9 May 2023
Published: 22 May 2023

1. Introduction

The exponential advances in artificial intelligence (AI) and machine learning (ML) have opened the door of automation to many applications. Examples are automatic speech recognition (ASR) [1] applied to personal assistants (e.g., SIRI® or Amazon's ALEXA®) and natural language processing (NLP) and understating [2] for different tasks such as sentiment analysis [3] and user intent detection [4]. Even though these advances are remarkable, many applications have lagged behind due to their critical matter, imperative near-to-perfect performance or simply because the users or administrators only trust the already existing legacy systems. One clear example is air traffic control (ATC) communications.

In ATC communications, ATCos are required to issue verbal commands to pilots in order to keep control and safety of a given area of airspace, although there are different means of communication, such as 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. These messages are displayed on a flight deck visual display.

Research targeted at understanding spoken ATC communications in the military domain can be traced back to the 1970s [5], late 1980s [6], and 1990s [7]. Recent projects are aiming at integrating AI-based tools into ATC processes by developing robust acoustic-based AI systems for transcribing dialogues. For instance, MALORCA [8,9], HAAWAAI [10] and ATCO2 [11,12]. These latest projects have shown mature-enough ASR and NLP systems that demonstrate potential for deployment in real-life operation control rooms. Other fields of work are voice activity detection (VAD), diarization [13] and ASR [14–16]. In addition, a few researchers have gone further by developing techniques to understand the ATCo–pilot dialogues [9,11,17]. However, previous works are mostly disentangled from each other. Some researchers only focus on ASR [18,19], while a few prior studies have integrated natural language understanding into their ASR pipelines [14,20].

Another key application that has seen growth in interest is the ATCo training framework. Training ATCos usually involves a human simulation-pilot. The simulation-pilot responds to or issues a request to the ATCo trainee in order to simulate an ATC communication with standard phraseology [21]. It is a human-intensive task, where a specialized workforce is needed during ATCo training and the overall cost is usually high. An example is the EUROCONTROL's ESCAPE lite simulator https://www.eurocontrol.int/simulator/escape, accessed on 12 May 2023) which still requires a human simulation-pilot. In a standard training scenario, the default simulation-pilots (humans) are required to execute the steps given by ATCo trainees, as in the case of real pilots (directly introduced to the simulator). The pilots, on the other hand, update the training simulator, so that the ATCos can see whether the pilots are following the desired orders. Therefore, this simulation is very close to a real ATCo–pilot communication. One well-known tool for ATCo training is Eurocontrol's ESCAPE simulator. It is an air traffic management (ATM) real-time simulation platform that supports: (i) airspace design for en-route and terminal maneuvering areas; (ii) the evaluation of new operational concepts and ATCo tools; (iii) pre-operational validation trials; and most importantly, (ii) the training of ATCos [22]. In this paper, we develop a virtual simulation-pilot engine that understands ATCo trainees' commands and possibly can replace current simulators based on human simulation-pilots. In practice, the proposed virtual simulation-pilot can handle simple ATC communications, e.g., the first phase of the ATCo trainee's training. Thus, humans are still required for more complex scenarios. Analogous efforts of developing a virtual simulation-pilot agent (or parts of it) have been covered in [23,24].

In this paper, we continue our previous work presented at SESAR Innovation Days 2022 [25]. There, a simple yet efficient 'proof-of-concept' virtual simulation-pilot was introduced. This paper formalizes the system with additional ATM-related modules. It also demonstrates that open-source AI-based models are a good fit for the ATC domain. Figure 1 contrasts the proposed pipeline (left side) and the current (default) human-based simulation-pilot (right side) approaches for ATCo training.

Main contributions Our work proposes a novel virtual simulation-pilot system based on fine-tuning several open-source AI models with ATC data. Our mains contributions are:

- Could human simulation pilots be replaced (or aided) by an autonomous AI-based system? This paper presents an end-to-end pipeline that utilizes a virtual simulation-pilot capable of replacing human simulation-pilots. Implementing this pipeline can speed up the training process of ATCos while decreasing the overall training costs.
- Is the proposed virtual simulation-pilot engine flexible enough to handle multiple ATC scenarios? The virtual simulation-pilot system is modular, allowing for a wide range of domain-specific contextual data to be incorporated, such as real-time air surveillance data, runway numbers, or sectors from the given training exercise. This flexibility boosts the system performance, while making its adaptation easier to various simulation scenarios, including different airports.

- Are open-source AI-based tools enough to develop a virtual simulation-pilot system? Our pipeline is built entirely on open-source and state-of-the-art pre-trained AI models that have been fine-tuned on the ATC domain. The Wav2Vec 2.0 and XLSR [26,27] models are used for ASR, BERT [28] is employed for natural language understanding (NLU), and FastSpeech2 [29] is used for the text-to-speech (TTS) module. To the best of our knowledge, this is the first study that utilizes open-source ATC resources exclusively [11,30–32].
- Which scenarios can a virtual simulation-pilot handle? The virtual simulation-pilot engine is highly versatile and can be customized to suit any potential use case. For example, the system can employ either a male or a female voice or simulate very high-frequency noise to mimic real-life ATCo–pilot dialogues. Additionally, new rules for NLP and ATC understanding can be integrated based on the target application, such as approach or tower control.

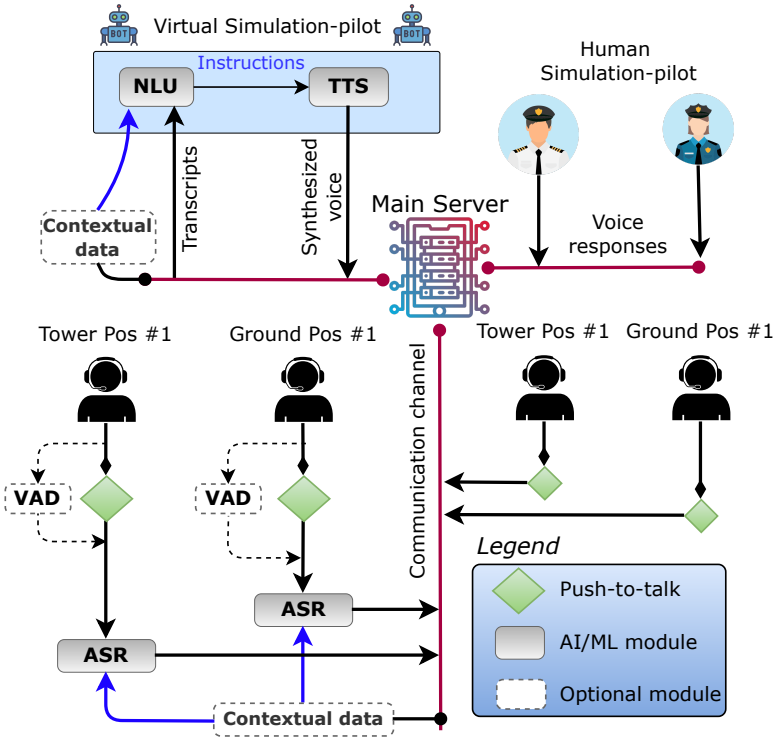

Figure 1. Virtual simulation-pilot pipeline for ATCo training. A traditional ATCo training setup is depicted on the right side, while our proposed virtual simulation-pilot is on the left side. The pipeline starts with an ATCo trainee issuing a communication and its capture after the end of the push-to-talk (PTT) signal, or voice-activity detection if not available. Then, the ASR and ***high-level entity parser*** (NLP) modules transcribe and extract the ATC-related entities from the voice communication. The output is later rephrased with simulation-pilot grammar. The speech synthesizer uses the generated text to create a WAV file containing the spoken textual prompt. In the end, a response is generated by the virtual simulation-pilot that matches the desired read-back.

The authors believe this research is a game changer in the ATM community due to two aspects. First, a novel modular system that can be adjusted to specific scenarios, e.g., aerodrome control or area control center. Second, it is demonstrated that open-source models such as XLSR [27] (for ASR) or BERT [28] (for NLP and ATC understanding) can be successfully adapted to the ATC scenario. In practice, the proposed virtual simulation-pilot engine could become the starting point to develop more inclusive and mature systems aimed at ATCo training.

The rest of the paper is organized as follows. Section 2 describes the virtual simulation-pilot system, covering the fundamental background for each of the base (Section 2.1) and

optional modules (Section 2.2) of the system. Section 3 describes the databases used. Then, Section 4 covers the experimental setup followed for adapting the virtual simulation-pilot and the results for each module of the system. Finally, brief future research directions are provided in Section 5 and the paper is concluded in Section 7.

2. Virtual Simulation-Pilot System

The virtual simulation-pilot system manages the most commonly used commands in ATC. It is particularly well-suited for the early stages of ATCo training. Its modular design allows the addition of more advanced rules and grammar to enhance the system's robustness. Our goal is to enhance the foundational knowledge and skills of ATCo trainees. Furthermore, the system can be customized to specific conditions or training scenarios, such as when the spoken language has a heavy accent (e.g., the case of foreign English) or when the ATCo trainee is practicing different positions.

In general, ATC communications play a critical role in ensuring the safe and efficient operation of aircraft. These communications are primarily led by ATCos, who are responsible for issuing commands and instructions to pilots in real-time. The training process of ATCos involves three stages: (i) initial, (ii) operational, and (iii) continuation training. The volume and complexity of these communications can vary greatly depending on factors such as the airspace conditions and seasonal fluctuations, with ATCos often facing increased workloads during peak travel seasons [25]. As such, ATCo trainees must be prepared to handle high-stress and complex airspace situations through a combination of intensive training and simulation exercises with human simulation-pilots [33]. In addition to mastering the technical aspects of air traffic control, ATCo trainees must also develop strong communication skills, as they are responsible for ensuring clear and precise communication with pilots at all times.

Due to the crucial aspect of ATC, efforts have been made to develop simulation interfaces for their training [33–35]. Previous works includes optimization of the training process [36], post-evaluation of each training scenario [37,38], and virtual simulation-pilot implementation, for example, a deep learning (DL)-based implementation [39]. In [24], the authors use sequence-to-sequence DL models to map from spoken ATC communications to high-level ATC entities. They use the well-known Transformer architecture [40]. Transformer is the base of the recent, well-known encoder–decoder models for ASR (Wav2Vec 2.0 [26]) and NLP (BERT [28]). The subsections address in more detail each module that is a part of the virtual simulation-pilot system.

2.1. Base Modules

The proposed virtual simulation-pilot system (see Figure 1) is built with a set of base modules, and possibly, optional submodules. The most simple version of the engine contains only the base modules.

2.1.1. Automatic Speech Recognition

Automatic speech recognition (ASR) or speech-to-text systems convert speech to text. An ASR system uses an acoustic model (AM) and a language model (LM). The AM represents the relationship between a speech signal and phonemes/linguistic units that make up speech and is trained using speech recordings along with their corresponding text. The LM provides a probability distribution over a sequence of words, provides context to distinguish between words and phrases that sound similar and is trained using a large corpus of text data. A decoding graph is built as a weighted finite state transducer (WFST) [41–43] using the AM and LM that generates text output given an observation sequence. Standard ASR systems rely on a lexicon, LM and AM, 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. The first is hybrid-based ASR, while the second is a more recent approach, termed end-to-end ASR. A comparison of both is shown in Figure 2.

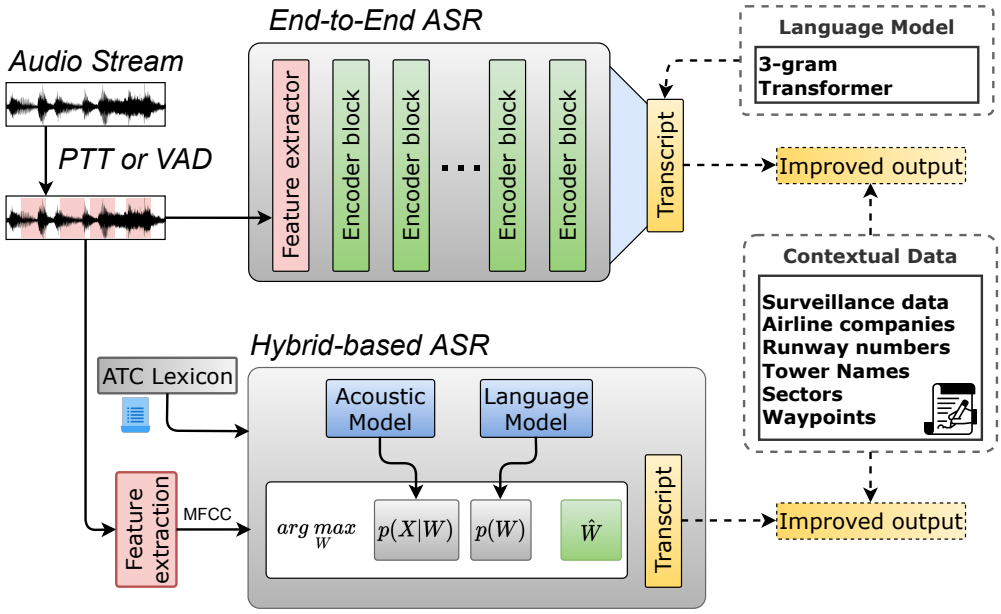

Figure 2. Traditional hybrid-based and more recent end-to-end automatic speech recognition systems. These systems take an ATCo voice communication as input and then produce transcripts as output. The dotted blocks refer to modules that are optional. For instance, surveillance data or other types of data (e.g., sector or waypoints) can be added to increase the overall performance of the system.

Hybrid-Based Automatic Speech Recognition. ASR with hybrid systems is based on hidden Markov models (HMM) and deep neural networks (DNN) [44]. DNNs are an effective module for 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 [45]. Hybrid systems remain one of the best approaches for building ASR engines based on lattice-free maximum mutual information (LF-MMI) [46]. Currently, HMM-DNN-based ASR is the state-of-the-art system for ASR in ATC domain [15].

Recent work in ASR has targeted different areas in ATC. For instance, a benchmark for ASR on ATC communications databases is established in [47]. Leveraging non-transcribed ATC audio data using semi-supervised learning has been covered in [48,49] and using self-supervised learning for ATC in [18]. The previous work related to the large-scale automatic collection of ATC audio data from different airports worldwide is covered in [15,50]. Additionally, innovative research aimed at improving callsign recognition by integrating surveillance data into the pipeline is covered by [10,12]. ASR systems are also employed for more high-level tasks such as pilot report extractions from very-high frequency (VHF) communications [51]. Finally, multilingual ASR has also been covered in ATC applications in [19].

The main components of a hybrid system are a pronunciation lexicon, LM and AM. 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. The system is composed of an explicit acoustic and language model. A visual example of hybrid-based ASR systems is in the bottom panel of Figure 2. Most of these systems can be trained with toolkits such as Kaldi [52] or Pkwrap [53].

End-to-End Automatic Speech Recognition. End-to-end (E2E) systems are based on a different paradigm compared to hybrid-based 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 a necessary separate component in standard hybrid-based systems. Unlike the hybrid approach, E2E models are learning a direct mapping between acoustic frames and model label units (characters, subwords or words) in one step toward the final objective of interest.

Recent work on encoder–decoder ASR can be categorized into two main approaches: connectionist temporal classification (CTC) [54] and attention-based encoder–decoder systems [55]. First, CTC uses intermediate label representation, allowing repetitions of labels and occurrences of 'blank output', which labels an output with 'no label'. Second, attention-based encoder–decoder or only-encoder models directly learn a mapping from the input acoustic frames to character sequences. For each time step, the model emits a character unit conditioned on the inputs and the history of the produced outputs. The important lines of work for E2E-ASR can be categorized as self-supervised learning [56–58] for speech representation, covering bidirectional models [26,59] and autoregressive models [60,61].

Moreover, recent innovative research on E2E-ASR for the ATC domain is covered in [62]. Here, the authors follow the practice of fine-tuning a Wav2Vec 2.0 model [26] with public and private ATC databases. This system reaches on-par performances with hybrid-based ASR models, demonstrating that this new paradigm for ASR development also performs well in the ATC domain. In E2E-ASR, the system encodes directly an acoustic and language model, and it produces transcripts in an E2E manner. A visual example of an only-encoder E2E-ASR system is in the top panel of Figure 2.

2.1.2. Natural Language Understanding

Natural language understanding (NLU) is a field of NLP that aims at reading comprehension. In the field of ATC, NLU is related to intent detection and slot filling. The slot-filling task is akin to named entity recognition (NER). In intent detection, the commands from the communication are extracted, while slot filling refers to the values of these commands and callsigns. Furthermore, throughout the paper, the system that extracts the high-level ATC-related knowledge from the ASR outputs is called a ***high-level entity parser*** system. The NER-based understanding of ATC communications has been previously studied in [11,23,24], while our earlier work [25] includes the integration of named-entity recognition (NER) into the virtual simulation-pilot framework.

The ***high-level entity parser*** system is responsible for identifying, categorizing and extracting crucial keywords and phrases from ATC communications. In NLP, these keywords are classified into pre-defined categories such as parts of speech tags, locations, organizations or individuals' names. In the context of ATC, the key entities include callsigns, commands and values (which includes units, e.g., flight level). For instance, consider the following transcribed communication (taken from Figure 3):

ASR transcript: ryanair nine two bravo quebec turn right heading zero nine zero,

would be parsed to high-level ATC entity format:

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

The previous output is then used for further processing tasks, e.g., generating a simulation-pilot-like response, metadata logging and reporting, or simply to help ATCos in their daily tasks. Thus, NLU is mostly focused on NER [63]. Initially, NER relied on the manual crafting of dictionaries and ontologies, which led to complexity and human error when scaling to more entities or adapting to a different domain [64]. The advancement of ML-based methods for text processing, including NER, has been introduced by [65]. The previous work [66] continued to advance NER techniques. A ***high-level entity parser*** system (such as ours) can be implemented by fine-tuning a pre-trained LM for the NER task. Currently, state-of-the-art NER models utilize pre-trained LMs such as BERT [28], RoBERTa [67] or DeBERTa [68]. For the proposed virtual simulation-pilot, we use a fine-tuned BERT on ATC text data.

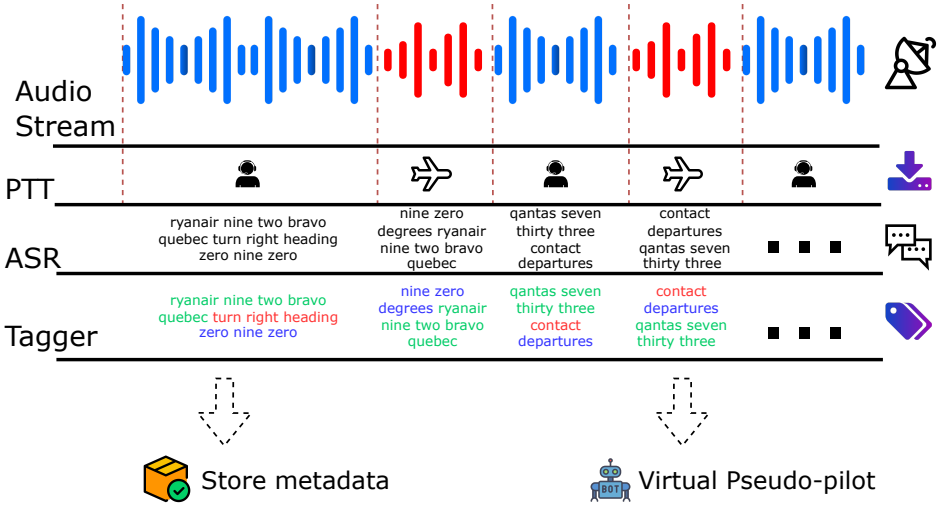

Figure 3. Detailed outputs of the main ML-based submodules of our proposed simulation-pilot system. It includes pre-processing from the input audio stream, speaker role detection by push-to-talk (PTT) signal, transcript generation and callsign/command/value extraction with the high-level entity with ASR and NER modules, respectively. All the data are later aggregated, packaged and sent to the response generator and TTS module. Note that these data can also be logged into a database for control and recording. Figure adapted from our previous work [25].

2.1.3. Response Generator

The response generator (RG) is a crucial component of the simulation-pilot agent. It processes the output from the ***high-level entity parser*** system, which includes the callsign, commands and values uttered by the ATCo, and then later generates a spoken response. The response is then delivered in the form of a WAV file, which is played through the headphones of the ATCo trainee. Additionally, the response, along with its metadata, can be stored for future reference and evaluation. The RG system is designed to generate responses that are grammatically consistent with what a standard simulation-pilot (or pilot) would say in response to the initial commands issued by the ATCo. The RG system comprises three submodules: (i) grammar conversion, (ii) a word fixer (e.g., ATCo-to-pilot phrase fixer), and (iii) text-to-speech, also known as a speech synthesizer. A visual representation of the RG system split by submodules is in Figure 4.

Grammar Conversion Submodule. A component designed to generate the response of the virtual simulation-pilot. First, the output of the ***high-level entity parser*** module (discussed in Section 2.1.2) is input to the grammar conversion submodule. At this stage, the communication knowledge has already been extracted, including the callsign, commands and their values. This is followed by a grammar-adjustment process, where the order of the high-level entities is rearranged. For example, we take into account the common practice of pilots mentioning the callsign at the end of the utterance while ATCos mention it at the beginning of the ATC communication. Thus, our goal is to align the grammar used by the simulation-pilot with the communication style used by the ATCo. See the first left panel in Figure 4.

Word Fixer Submodule. This is a crucial component of the virtual simulation-pilot system that ensures that the output from the response generator aligns with the standard ICAO phraseology. This is achieved by modifying the commands to match the desired response based on the input communication from the ATCo. The submodule applies specific mapping rules, such as converting *descend* $\rightarrow$ *descending* or *turn* $\rightarrow$ *heading*, to make the generated reply as close to standard phraseology as possible. Similar efforts have been covered in a recent study [39] where the authors propose a *copy mechanism* that copies the key entities from the ATCo communication into the desired response of the virtual simulation-pilot, e.g., *maintain* $\rightarrow$ *maintaining*. In real-life ATC communication, however,

the wording of ATCos and pilots slightly differs. Currently, our *word fixer* submodule contains a list of 18 commands but can be easily updated by adding additional mapping rules to a `rules.txt` file. This allows the system to adapt to different environments, such as aerodrome control, departure/approach control or area control center. The main conversion rules used by the word fixer submodule are listed in Table 1. The ability to modify and adapt the word fixer submodule makes it a versatile tool for training ATCos to recognize and respond to standard ICAO phraseology. See the central panel in Figure 4.

Text-to-Speech Submodule. Speech synthesis, also referred to as text-to-speech (TTS), is a multidisciplinary field that combines various areas of research such as linguistics, speech signal processing and acoustics. The primary objective of TTS is to convert text into an intelligible speech signal. Over the years, numerous approaches have been developed to achieve this goal, including formant-based parametric synthesis [69], waveform concatenation [70] and statistical parametric speech synthesis [71]. In recent times, the advent of deep learning has revolutionized the field of TTS. Models such as Tacotron [72] and Tacotron2 [73] are end-to-end generative TTS systems that can synthesize speech directly from text input (e.g., characters or words). Most recently, FastSpeech2 [29] has gained widespread recognition in the TTS community due to its simplicity and efficient non-autoregressive manner of operation. Finally, TTS is a complex field that draws on a variety of areas of research and has made significant strides recently, especially with the advent of deep learning. For a more in-depth understanding of the technical aspects of TTS engines, readers are redirected to [74] and novel diffusion-based TTS systems in [75]. The TTS system for ATC is depicted in the right panel in Figure 4.

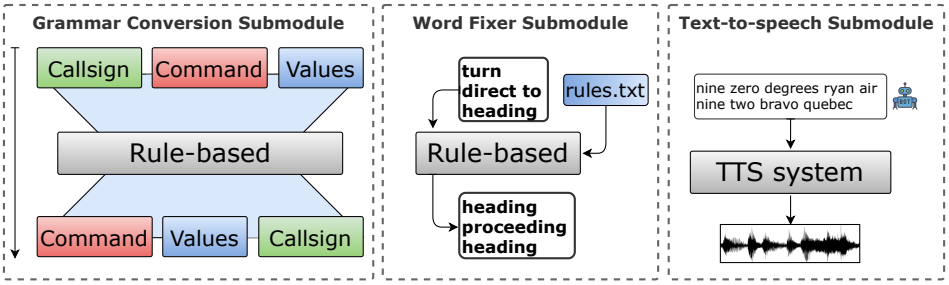

Figure 4. Detailed submodules of the response generator.

Table 1. Word-fixing rules. The rules are used to convert ATCos input communications into a virtual simulation-pilot response.

Word Fixer Submodule—Rules.txt	
Horizontal commands	**Handover commands**
continue heading → continuing altitude	contact tower → contact tower
heading → heading	station radar → station radar
turn → heading	squawk → squawk
turn by → heading	squawking → squawk
direct to → proceeding direct	contact frequency → NONE
Level commands	**Speed commands**
maintain altitude → maintaining altitude	reduce → reducing
maintain altitude → maintain	maintain speed → maintaining
descend → descending	reduce speed → reduce speed
climb → climbing	speed → NONE

2.2. Optional Modules

In contrast to the base modules, covered in Section 2.1, the optional modules are blocks that can be integrated into the virtual simulation-pilot to enhance or add new capabilities.

An example is the PTT (push-to-talk) signal. In some cases a PTT signal is not available; thus, voice activity detection can be integrated. Below, each of the proposed optional modules is covered in more detail.

2.2.1. Voice Activity Detection

Voice activity detection (VAD) is an essential component in standard speech-processing systems to determine which portions of an audio signal correspond to speech and which are non-speech, i.e., background noise or silence. VAD can be used for offline purpose decoding, as well as for online streaming recognition. The offline VAD is used to split a lengthy audio into shorter segments that can then be used for training or evaluating ASR or NLU systems. The online VAD is particularly crucial for ATC ASR when the PTT signal is not available. An example of an online VAD is the WebRTC (developed by Google https://webrtc.org/, accessed on 12 May 2023). In ATC communications, VAD is used to filter out the background noise and keep only the speech segments that carry the ATCo's (or pilot's) voice messages. One of the challenges for VAD in ATC communications is the presence of a high level of background noise. The noise comes from various sources, e.g., the engines of aircraft, wind or even other ATCos. ATC communications can easily have signal-to-noise (SNR) ratios lower than 15 dB. If VAD is not applied (and there is not a PTT signal available), the ASR system may degrade the accuracy of speech transcription, which may result in incorrect responses from the virtual simulation-pilot agent.

VAD has been explored before in the framework of ATC [76]. A general overview of recent VAD architecture and research directions is covered in [77]. Some other researchers have targeted how to personalize VAD systems [78] and how this module plays its role in the framework of diarization [79]. There are several techniques used for VAD, ranging from traditional feature-based models to hidden Markov models to Gaussian mixture-based models [80]. On the other hand, machine-learning-based models have proven to be more accurate and robust, particularly deep neural network-based methods. These techniques can learn complex relationships between the audio signal and speech and can be trained on large annotated datasets. For instance, convolutional and deep-neural-network-based VAD has received much interest [76]. VAD can be used in various stages of the ATC communication pipeline. VAD can be applied at the front-end of the ASR system to pre-process the audio signal and reduce the processing time of the ASR system. Figures 1 and 2 depict where a VAD module can be integrated into the virtual simulation-pilot agent.

2.2.2. Contextual Biasing with Surveillance Data

In order to enhance the accuracy of an ASR system's predictions, it is possible to use additional context information along with speech input. In the ATC field, radar data can serve as context information, providing a list of unique identifiers for aircraft in the airspace called "callsigns". By utilizing these radar data, the ASR system can prioritize the recognition of these registered callsigns, increasing the likelihood of correct identification. Callsigns are typically a combination of letters, digits and an airline name, which are translated into speech as a sequence of words. The lattice, or prediction graph, can be adjusted during decoding by weighting the target word sequences using the finite state transducer (FST) operation of composition [12]. This process, called lattice rescoring, has been found to improve the recognition accuracy, particularly for callsigns. Multiple experiments using ATC data have demonstrated the effectiveness of this method, especially in improving the accuracy of callsign recognition. The results of contextual biasing are presented and discussed below in Section 4.1.

Re-ranking module based on Levenshtein distance. The ***high-level entity parser*** system for NER (see Section 2.1.2) allows us to extract the callsign from a given transcript or ASR 1-best hypotheses. Recognition of this entity is crucial where a single error produced by the ASR system affects the whole entity (normally composed of three to eight words). Additionally, speakers regularly shorten callsigns in the conversation, making it impossible for an ASR system to generate the full entity (e.g., *'three nine two papa'* instead of *'austrian*

three nine two papa', *'six lima yankee'* instead of *'hansa six lima yankee'*). One way to overcome this issue is to re-rank the entities extracted by the ***high-level entity parser*** system with the surveillance data. The output of this system is a list of tags that match words or sequences of words in an input utterance. As our only available source of contextual knowledge is callsigns registered at a certain time and location, we extract callsigns with the ***high-level entity parser*** system and discard other entities. Correspondingly, each utterance has a list of callsigns expanded into word sequences. As input, the re-ranking module takes (i) a callsign extracted by the ***high-level entity parser*** system and (ii) an expanded list of callsigns. The re-ranking module compares a given n-gram sequence against a list of possible n-grams and finds the closest match from the list of surveillance data based on the *weighted Levenshtein distance*. In order to use contextual knowledge, it is necessary to know which words in an utterance correspond to a desired entity (i.e., a callsign), which is why it is necessary to add into the pipeline the ***high-level entity parser*** system. We skip the re-ranking in case the output is a 'NO_CALLSIGN' flag (no callsign recognized).

2.2.3. Read-Back Error-Insertion Module

The approach of using the virtual simulation-pilot system can be adapted to meet various communication requirements in ATC training. This includes creating a desirable read-back error (RBE), which is a plausible scenario in ATC, where a pilot or ATCo misreads or misunderstands a message [81]. By incorporating this scenario in ATCos' training, they can develop the critical skills for spotting these errors. This is a fundamental aspect of ensuring the safety and efficiency of ATM [82]. The ability to simulate (by inserting a desired error) and practice these scenarios through the use of the virtual simulation-pilot system offers a valuable tool for ATCo training and can help to improve the overall performance of ATC. An example could look like: `ATCo: turn right` → `Pilot (RBE): turning left`.

The structure of the generated RBE could depend on the status of the exercise, for instance, whether the ATCo trainee is in the aerodrome control or approach/departure control position. These positions should, in the end, change the behavior of this optional module. The proposed, optional RBE insertion module is depicted in Figure 5.

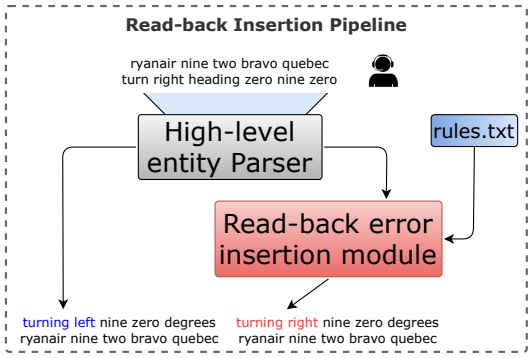

Figure 5. Read-back insertion module. At first, an input transcribed communication is sent to the ***high-level entity parser*** in order to extract the knowledge, i.e., callsign, commands and values. Later, with a defined probability, a desired read-back error can be inserted

3. Datasets

This section describes the public databases used for training and evaluating the different modules of our virtual simulation-pilot system. In addition, Table 2 summarizes well-known private and public ATC-related databases [83]. Our goal is to conduct a thorough and comprehensive study on the use of virtual simulation-pilots in ATC. To ensure the reproducibility of our research, we use either only open-source databases or a combination of these and private databases during the training phase of our base models. The exception is the TTS module. The TTS system is a pre-trained out-of-the-box module downloaded from HuggingFace (the TTS is part of the response generator). Despite this, we aim at demonstrating the potential of the virtual simulation pilot in a more realistic

setting. Hence, the system is also evaluated on highly challenging private databases that the authors have access to. These databases cover real-life ATC communications, which might contain high levels of background or cockpit noise. The results achieved in this work can provide a better idea of the performance of our approach in practical applications, while also highlighting its strengths and weaknesses in a real-world scenario. In any case, our focus remains on ensuring that our research is thoroughly documented and that it can be easily replicated by other researchers in the ATC and ATM field.

3.1. Public Databases

LDC-ATCC corpus: The air traffic control corpus (ATCC) (available for download in: https://catalog.ldc.upenn.edu/LDC94S14A, accessed on 12 May 2023) consists of recorded speech initially designed for research on ASR. Here, the metadata is also used for NLU research, e.g., speaker role detection. The audio data contain 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. The metadata cover voice activity segmentation details, speaker role information (who is talking) and callsigns in ICAO format. In addition, the corpus consists of approximately 25 h of ATCo and pilot transmissions (after VAD).

UWB-ATCC corpus: The UWB-ATCC corpus (released by the University of West Bohemia, see: https://lindat.mff.cuni.cz/repository/xmlui/handle/11858/00-097C-0000-0001-CCA1-0, accessed on 12 May 2023) 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 h. The audio data are mono-channel sampled at 8 kHz and 16-bit PCM.

ATCO2 corpus: The dataset was built for the development and evaluation of ASR and NLP technologies for English ATC communications. The dataset consists of English coming from several airports worldwide (e.g., LKTB, LKPR, LZIB, LSGS, LSZH, LSZB, YSSY). We used this corpus twofold. First, we employed up to 2500 h of audio data of the official pseudo-annotated set (see more information in [11]) for training our ASR systems; this training set is labeled *ATCO2-PL set corpus*. It is worth mentioning that the transcripts of the ATCO2 training corpus were automatically generated by an ASR system. Despite this, recent work has shown its potential to develop robust ASR systems for ATC from scratch, e.g., [11]. Second, for completeness, we use the two official partitions of the ATCO2 test set, namely, the *ATCO2 test set 1h corpus* and the *ATCO2 test set 4h corpus*, as evaluation sets. The first corpus contains 1.1 h of open-source transcribed annotations, and it can be accessed for free at: https://www.atco2.org/data (accessed on 12 May 2023). The latter contains ∼3 h of extra annotated data, and the full corpus is available for purchase through ELDA at: http://catalog.elra.info/en-us/repository/browse/ELRA-S0484 (accessed on 12 May 2023). The recordings are mono-channel sampled at 16 kHz and 16-bit PCM.

ATCOSIM corpus: This is a free public database for research on ATC communications. It comprises 10 h of speech data recorded during real-time ATC simulations using a close-talk headset microphone. The utterances are in the English language and pronounced by ten non-native speakers. The speakers are split by gender. Even though we do not pursue this direction, the ATCOSIM corpus can be used for the development or adaptation of already-existing TTS systems to ATC with voices from different genders, i.e., males or females. ATCOSIM also includes orthographic transcriptions and additional information about the speakers and recording sessions [32]. This dataset can be accessed for free at: https://www.spsc.tugraz.at/databases-and-tools (accessed on 12 May 2023).

Table 2. Air traffic control communications-related databases. ⋆ abbreviation in IETF format. † research directions that can be explored based on the annotations provided by the dataset. †† ASR and TTS are interchangeable; the same annotations of each recording can be used to fine-tune or train a TTS module. § denotes datasets that contain annotations on the callsign or/and command level. SpkID: Speaker role identification.

Characteristics			Research Topics †			Other	
Database	**Accents** ⋆	**Hrs**	**ASR/TTS** ††	**SpkID**	**NLU** §	**License**	**Ref.**
Private databases							
MALORCA	cs, de	14	✓	✓	✓	✗	[48]
AIRBUS	fr	100	✓	-	✓	✗	[83]
HAAWAII	is, en-GB	47	✓	✓	✓	✗	[62]
Internal	several	44	✓	✗	✗	✗	-
Public databases							
ATCOSIM	de, fr, de-CH	10.7	✓	✗	✗	✓	[32]
UWB-ATCC	cs	13.2	✓	✓	✗	✓	[31]
LDC-ATCC	en-US	26.2	✓	✓	✓	✓	[30]
HIWIRE	fr, it, es, el	28.7	✓	✗	✗	✓	[84]
ATCO2	several	5285	✓	✓	✓	✓	[11]

3.2. Private Databases

MALORCA corpus: MAchine Learning Of speech Recognition models for Controller Assistance: http://www.malorca-project.de/wp/ (accessed on 12 May 2023). This dataset is based on a research project that focuses to propose a general, cheap and effective solution to develop and automate speech recognition for controllers using the speech data and contextual information. The data collected are mainly from the Prague and Vienna airports, which is around 14 h. The data are split into training and testing sets with a split amount of 10 h and 4 h (2 h from each airport), respectively.

HAAWAII corpus: Highly Advanced Air Traffic Controller Workstation with Artificial Intelligence Integration: https://www.haawaii.de (accessed on 12 May 2023): This dataset is 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 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 training and 4 h for testing). The 4 h test set is taken—2 h each—from both London and Iceland airports. Similar to another corpus, the audio files are sampled at 8 kHz and 16-bit PCM. This corpus is only used as an out-of-domain dataset; thus, we only report the results on the ASR level.

Internal data: In addition to the above-mentioned datasets, we have data from some industrial research projects that amount to a total duration of 44 h of speech recordings of ATCos and pilots along with their manual transcripts.

4. Experimental Setup and Results

In this section, we present the experimental results for some modules described in Section 2. These modules are trained with the datasets from Section 3. Note that not all datasets are used during the training and testing phases.

4.1. Automatic Speech Recognition

This subsection list the results related to ASR, previously covered in Section 2.1.1. We analyze (i) the three proposed ASR architectures, (ii) the training datasets used during the

training phase, and (iii) the experimental setup and results obtained on different public and private test sets.

4.1.1. Architectures

The results of ASR are split in two. First, we evaluate hybrid-based ASR models, which are the default in current ATC-ASR research [8,9]. Second, we train ASR models with state-of-the-art end-to-end architectures, e.g., Transformer-based [27,40] and Conformer-based [85]. The experimental setup and results analysis (below) for each proposed model refers to the results from Table 3.

Hybrid-based ASR: For the hybrid-based ASR experiments, we use conventional biphone convolutional neural network (CNN) [86] + TDNN-F [46]-based acoustic models trained with the Kaldi [52] toolkit (i.e., nnet3 model architecture). The AMs are trained with the LF-MMI training framework, considered to produce a state-of-the-art performance for hybrid ASR. In all the experiments, 3-fold speed perturbation with MFCCs and i-vector features is used. The LM is trained as a statistical 3-gram model using manual transcripts. Previous work related to ATC with this architecture is in [11,15].

XLSR-KALDI ASR: In [87], the authors propose to use the LF-MMI criterion (similar to hybrid-based ASR) for the supervised adaptation of the self-supervised pre-trained XLSR model [27]. They also show that this approach outperforms the models trained with only the supervised data. Following that technique, we use the XLSR [27] model pre-trained with a dataset as large as 50 k h of speech data, and later we fine-tune it with the supervised ATC data using the LF-MMI criterion. Further details about the architecture and experimental setup for pre-training the XLSR model can be found in the original paper [27]. The results for this model are in the row tagged as '`XLSR-KALDI`' in Table 3.

End-to-End ASR: We use the SpeechBrain [88] toolkit to train a Conformer [85] ASR model with ATC audio data. The Conformer model is composed of 12 encoder layers and an additional 4 decoder layers (transformer-based [40]). We reuse the `Conformer-small` recipe from LibriSpeech [89] and adapt it to the ATC domain. See the recipe at: https://github.com/speechbrain/speechbrain/tree/develop/recipes/LibriSpeech/ASR/transformer (accessed on 12 May 2023). The dimension of the encoder and decoder model is set to $d_{model} = 144$ with $d_{ffn} = d_{model} * 4$. This accounts for a total of 11M parameters. We use dropout [90] with a probability of $dp = 0.1$ for the attention and hidden layers, while Gaussian error linear units (GELU) is used as the activation function [91]. We use the Adam [92] optimizer with an initial learning rate of $\gamma = 1e{-}3$. We also use the default dynamic batching, which speeds up the training. During training, we combine the per-frame conformer decoder output and CTC probabilities [93]. The CTC loss [94] is weighted by $\alpha = 0.3$. During inference and evaluation, the beam size is set to 66 with a CTC weight of $ctc_w = 0.4$.

4.1.2. Training and Test Data

Training data configuration: To see the effectiveness of using automatically transcribed data, as well as comparing the performance on the in-domain VS out-of-domain sets, we train both the hybrid (CNN-TDNNF) and E2E (Conformer) models twice. First, we employ a mix between public and private supervised (recordings with gold annotations) ATC resources, which comprises around 190 h. We tag these models as ***scenario (a)—only supervised data***. Second, we use a subset of 500 h of pseudo-annotated recordings (a seed ASR system is used to transcribe the ATC communications from different airports) from the open-source ATCO2-PL set corpus (see introductory paper [11]). We tag this model as ***scenario (b)—only ATCO2-PL 500 h data***. The results referencing both scenarios are in Tables 3 and 4.

Test data configuration: Six different test sets are used for ASR evaluation, as shown in Table 3. The first four test sets (highlighted in orange) are private and the last two test sets (highlighted in blue) are open data. Each two consecutive test sets are taken from one project: (i) NATS and ISAVIA are part of the HAAWAII corpus, (ii) Prague and Vienna are

part of the MALORCA corpus, and (iii) ATCO2-1h and ATCO2-4h are from the ATCO2 project. Each dataset, along with the test split, is described in Section 3. We aimed at evaluating how the model's architecture, training paradigm (hybrid-based and E2E-ASR) and training data directly affect the performance of ASR.

4.1.3. Evaluation Metric

The preeminent technique for evaluating the efficacy of an ASR system is the word error rate (WER). This metric entails a meticulous comparison between the transcription of an utterance and the word sequence hypothesized by the ASR model. The WER is determined by computing the aggregate of three types of errors, specifically, substitutions (S), insertions (I) and deletions (D), over the total count of words within the transcription. Should the reference transcript comprise N words, the WER can be computed using Equation (1), outlined below.

$$WER = \frac{I + D + S}{N} \times 100. \quad (1)$$

We evaluate all the models from Tables 3 and 4 with WERs. For the boosting experiments (see Section 2.2.2 and Table 4) we additionally use *EntWER*, which evaluates WER only on the callsign word sequence, and *ACC*, which evaluates the accuracy of the system in capturing the target callsign in ICAO format.

4.1.4. Speech Recognition Results

The results of all the compared ASR models are in Table 3.

CNN-TDNNF model: This is our default architecture, as it has already been shown to work properly on the ATC domain. It has also been used largely in prior ATC work, such as ATCO2, HAAWAII and MALORCA (see Section 1). In our experiments, we trained this model for both *scenario (a)* and *scenario (b)*. Not surprisingly, we noted that the WERs are heavily impacted by the training and testing data. If we compare CNN-TDNNF scenario (a) VS scenario (b), we see a systematic drop in performance for NATS (7.5% WER → 26.7% WER) and ISAVIA (12.4% WER → 34.1% WER). However, in Prague and Vienna, which are still out-of-domain for scenario (b), less degradation in WERs is seen: 6.6% WER → 11.7% WER for Prague and 6.3% WER → 11.8% WER for Vienna.

XLSR-KALDI model: As mentioned earlier, we fine-tune the XLSR model (pre-trained based on wav2vec 2.0 [26]) with the supervised data from *scenario (a)*. We do this as a proof of concept. The results show that the performance is consistent over all the private test sets compared to the CNN-TDNNF model trained with the same data. Though the model has not seen the noisy ATCO2 data during fine-tuning, since this model is pre-trained with large amounts of data, the WER on the ATCO2 test sets significantly improves compared to the CNN-TDNNF model. We see an absolute improvement of 9.4% (27.4% → 18%) and 10.9% (36.6% → 25.7%) for the ATCO2-1h and ATCO2-4h test sets, respectively.

Conformer model: We evaluate Conformer [85], an encoder–decoder Transformer-based [40] model. With the Conformer architecture, we again train two models: on supervised data (*scenario (a)*) and on the 500 h ATCO2-PL set (*scenario (b)*). Both are tagged as *CONFORMER* in Table 3. Likewise, for CNN-TDNNF models, the first four test sets are deemed in-domain for the baseline model, whereas the last two test sets (ATCO2-1h and ATCO2-4h test sets) are considered out-of-domain. Conversely, the second model is optimized for the out-of-domain test sets, while the first four are considered out-of-domain. Our goal is to demonstrate the effectiveness of the *ATCO2-PL* dataset as an optimal resource for training models when only limited in-domain data are available. The second model demonstrates competitive performance when tested on close-mic speech datasets such as Prague and Vienna, which exclusively use the ATCo recordings. Yet, the model's performance deteriorates on more complex datasets, such as NATS and ISAVIA, which include pilot speech. We also note significant improvements on the ATCO2-1h and ATCO2-4h test sets when training with the ATCO2-PL dataset. Scenario (b) exhibits a 62% and 48% relative

WER reduction compared to scenario (a) on ATCO2-1h and ATCO2-4h, respectively. In contrast, the first model performed poorly on both: 41.8 and 46.2% WER, respectively. A critical consideration arises when examining the performance of the Conformer and CNN-TDNNF models under the same training scenario, scenario (b). Notably, the Conformer model outperforms the CNN-TDNNF model across all datasets, except for the Vienna test set. This leads us to hypothesize that the Conformer architecture shows greater proficiency when being trained over extensive datasets when compared to the CNN-TDNNF model in this particular scenario.

Table 3. WER for various public and private test sets with different ASR engines. The top results per block are **highlighted in bold**. The best result per test set is marked with an underline. ‡ datasets from HAAWAII corpus and †datasets from MALORCA project [8].

Model	Test Sets					
	NATS ‡	ISAVIA ‡	Prague †	Vienna †	ATCO2-1h	ATCO2-4h
	scenario (a)—only supervised data					
CNN-TDNNF	7.5	12.4	6.6	6.3	27.4	36.6
XLSR-KALDI	**7.1**	**12.0**	6.7	**5.5**	**18.0**	**25.7**
CONFORMER	9.5	13.7	**5.7**	7.0	41.8	46.2
	scenario (b)—only ATCO2-PL 500 h data					
CNN-TDNNF	26.7	34.1	11.7	**11.8**	19.1	25.1
CONFORMER	**21.6**	**32.5**	**7.6**	12.5	**15.9**	**24.0**

Callsign boosting with surveillance data (≈contextual biasing): The contextual biasing approach is introduced in Section 2.2.2. Table 4 demonstrates the effect of callsign boosting on the NATS test set (part of HAAWAII). The results of two ASR models are compared. Both models have the same architecture (Kaldi CNN-TDNNf) but are trained on different data. The first scenario (a), as in the experiments above, is trained on a combination of open-source and private annotated ATC databases that includes in-domain data (NATS); the second scenario (b) is trained on the 500 hours of automatically transcribed data collected and prepared for the ATCO2 project, which is out-of-domain data. As expected, the in-domain model performs better on the NATS dataset. At the same time, for both models, we can see a considerable improvement when contextual biasing is applied. The best results are achieved when only a ground-truth callsign is boosted, i.e., 86.7% → 96.1% ACC for scenario (a) and 39.9% → 70.0% ACC for scenario (b). As in real life, we usually do not have the ground-truth information, the improvement we can realistically obtain with radar data is shown in the line tagged as **N-grams**. The effectiveness of biasing also depends on the number of callsigns used to build the biasing FST, as the more false callsigns are boosted the noisier the final rescoring is. According to previous findings, the ideal size of the biasing FST for improving performance depends on the data, but typically, the performance begins to decline when there are more than 1000 contextual entities [95]. In our data, we have an average of 200 contextual entities per spoken phrase. For the n-gram-boosting experiments, we achieved a relative improvement in callsign WERs of 51.2% and 34% for callsign recognition with models (a) and (b), and 9.5% and 12.4% for the entire utterance, respectively (see Table 4).

Table 4. Results for boosting on NATS test set corpus (HAAWAII). We ablate two models: scenario (a), a general ATC model trained only on supervised data, and scenario (b), a model trained on the ATCO2-PL 500 h set. Results are obtained with offline CPU decoding. ¶ word error rates only on the sequence of words that compose the callsign in the utterance.

	General ATC Model			ATCO2 Model-500h		
Boosting	**WER**	**EntWER ¶**	**ACC**	**WER**	**EntWER ¶**	**ACC**
	scenario (a)—only supervised data			**scenario (b)—only ATCO2-PL 500 h data**		
Baseline	7.4	4.1	86.7	26.7	30.0	39.9
Unigrams	7.4	3.6	88.0	25.6	24.1	46.2
N-grams	6.7	2.0	93.3	23.4	19.8	61.3
GT boosted	6.4	1.3	96.1	22.0	16.2	70.0

4.2. High-Level Entity Parser

A NER system is trained to parse text into high-level entities relevant to ATC communications. The NER module (or tagger) is depicted in Figure 3. First, a BERT [28] model is downloaded from HuggingFace [96,97] which is then fine-tuned on the NER task with 3k sentences (∼3 h of speech) using the *ATCO2 test set corpus* (the pre-trained version of `BERT-base-uncased` with 110 million parameters is used, see at: https://huggingface.co/bert-base-uncased, accessed on 12 May 2023). In this corpus, each word has a tag that corresponds to either callsign, command, values or UNK (everything else). The final layer of the BERT model is replaced by a linear layer with a dimension of 8 (this setup follows the class structures from Section 3.3 of the paper: [13], i.e., two outputs for each class). As only 3k sentences are used, a 5-fold cross-validation is conducted to avoid overfitting. Further details about experimentation are covered in [10]. We redirect the reader to the public and open-source GitHub repository of the ATCO2 corpus (ATCO2 GitHub repository: https://github.com/idiap/atco2-corpus, accessed on 12 May 2023).

Experimental setup: we fine-tune each model on an NVIDIA GeForce RTX 3090 for 10k steps. During experimentation, we use the same learning rate of $\gamma = 5e{-}5$, with a linear learning rate scheduler. The dropout [90] is set to $dp = 0.1$ for the attention and hidden layers, while GELU is used as an activation function [91]. We also employ gradient norm clipping [98]. We fine-tune each model with an effective batch size of 32 for 50 epochs with the AdamW optimizer [99] ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon$ =1e−8).

Evaluation metric: we evaluate our NER system with the F-score. The F-score or F-measure is a statistical measure utilized in binary classification analysis to evaluate a test's accuracy. The F1-score, defined in Equation (4), represents the harmonic mean of precision and recall. Precision, as described in Equation (2), is the ratio of true positive (TP) results to all positive results (including false positives (FP)), while recall, as defined in Equation (3), is the ratio of TP to all samples that should have been identified as positive (including false negatives (FN)):

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{4}$$

Results: the ***high-level entity parser*** system is evaluated on the only available public resource, the ATCO2-4h test set, which contains word-level tags, i.e., callsign, command and values. The results for precision, recall and F1-score over each of the proposed classes are listed in Table 5. Our BERT-based system achieves a high level of performance in callsign detection, with an F1-score of 97.5%. However, the command and values classes show

an average worse performance, with F1-scores of 82.0% and 87.2%, respectively. Notably, the command class presents the greatest challenge due to its inherent complexity when compared to values and callsigns. Values are predominantly composed of keywords, such as "flight level" followed by cardinal numbers such as "two", "three hundred" or "one thousand". These characteristics make them easy for a system to recognize. Similarly, callsigns are highly structured, consisting of an airline designator accompanied by numbers and letters spoken in the radiotelephony alphabet [21]. Given their importance in communication, as in callsign highlighting [11] or read-back error detection [81], additional validation is necessary for real-life scenarios or when working with proprietary/private data.

Although our BERT-based system achieves a high performance in callsign recognition, there is still room for improvement. One potential method for enhancing the performance is to incorporate real-time surveillance data into the system [10], which was introduced in Section 2.2.2 and Table 4.

4.3. Response Generator

The response generator is composed of three submodules (see Section 2.1.3). The **grammar conversion** and **word fixer** submodules are based on hard-coded rules. Thus, quantitative results are not provided.

4.4. Text to Speech

Text-to-speech (TTS) is a technology that facilitates the conversion of written text into spoken language. When employed in ATC, TTS can be integrated with virtual simulation-pilot systems to train ATCos. In this study, a state-of-the-art non-autoregressive speech synthesizer, the FastSpeech2 model [29], is utilized. A pre-trained TTS model is downloaded from the HuggingFace hub [96]. Access to the model can be obtained at https://huggingface.co/facebook/fastspeech2-en-ljspeech, accessed on 12 May 2023. FastSpeech2 is used in inference mode with the sentence produced by the grammar conversion submodule, and subsequently, the word fixer submodule, serving as the prompt of the virtual simulation-pilot. Other models, such as Tacotron [72] or Tacotron2 [73] (free access to the Tacotron2 model can be found at https://huggingface.co/speechbrain/tts-tacotron2-ljspeech, accessed on 12 May 2023), can be fine-tuned and implemented to handle ATC data.

System analysis: In our experiments, we discovered that the model is capable of handling complex word sequences, such as those commonly encountered in ATC, including read-backs from virtual simulation-pilots that contain multiple commands and values. However, we did not conduct any qualitative analysis of the TTS-produced voice or speech, leaving this as a future area of exploration. We also did not investigate the possibility of fine-tuning the TTS module with ATC audio data, as our main focus was on developing a simple and effective virtual simulation-pilot system using pre-existing open-source models.

Future lines of work: Although we did not pursue this area, it is indeed possible to fine-tune the TTS module using in-domain ATC data. In Table 2, we provide a list of both public and private databases that could be utilized for this purpose. Generally, the same annotations used for ASR can also be applied to fine-tune a TTS system. However, there are two different approaches that could be explored simultaneously or sequentially. First, by considering the speaker role (ATCo or pilot), the TTS module could be biased to produce speech that is more appropriate for different roles, such as noisy speech from pilots. Second, if datasets are available that provide information on gender and accents, such as the ATCOSIM dataset [32], TTS models with different accents and gender could be developed.

Table 5. F1-score (@F1), precision (@P) and recall (@R) metrics for callsign, command and value classes of the ***high-level entity parser*** system. Results are averaged over a 5-fold cross-validation scheme on the *ATCO2-4h corpus* in order to mitigate overfitting. We run fine-tuning five times with different training seeds (2222/3333/4444/5555/6666).

Model	Callsign			Command			Values		
	@P	@R	@F1	@P	@R	@F1	@P	@R	@F1
`bert-base-uncased`	97.1	97.8	97.5	80.4	83.6	82.0	86.3	88.1	87.2

5. Limitations and Future Work

In our investigation of ASR systems, we have explored the potential of hybrid-based and E2E ASR systems, which can be further enhanced by incorporating data relevant to the specific exercise undertaken by ATCo trainees, such as runway numbers or waypoint lists. Moving forward, we suggest that research should continue to explore E2E training techniques for ASR, as well as methods for integrating contextual data into these E2E systems.

The repetition generator currently in use employs a simple grammar converter and a pre-trained TTS system. However, we believe that additional efforts could be made to enhance the system's ability to convey more complex ATC communications to virtual simulation-pilots. In particular, the TTS system could be fine-tuned to produce female or male voices, as well as modify key features such as the speech rate, noise artifacts or cues to synthesize voices in a stressful situation. Additionally, a quantitative metric for evaluating the TTS system could be integrated to further enhance its efficacy. We also list some optional modules (see Section 2.2) that can be further explored, e.g., the read-back insertion module or voice activity detection.

Similarly, there is scope for the development of multimodal and multitask systems. Such systems would be fed with real-time ATC communications and contextual data simultaneously, later generating transcripts and high-level entities as the output. Such systems could be considered a dual ASR and high-level entity parser. Finally, the legal and ethical challenges of using ATC audio data are another important field that needs to be further explored in future work. We redirect the reader to the **legal and privacy aspects for collection of ATC recordings** section in [11].

6. How to Develop Your Own Virtual Simulation-Pilot

If you would like to replicate this work with in-domain data, i.e., for a specific scenario or airport, you can follow the steps below:

1. Start by defining the set of rules and grammar to use for the annotation protocol. You can follow the cheat-sheet from the ATCO2 project [11]. See https://www.spokendata.com/atc and https://www.atco2.org/, accessed on 12 May 2023. In addition, one can use previous ontologies developed for ATC [17].
2. For training or adapting the virtual simulation-pilot engine, you need three sets of annotations: (i) gold annotations of the ATCo–pilot communications for ASR adaptation; (ii) high-level entity annotations (callsign, command and values) to perform NLU for ATC; and (iii) a set of rules to convert ATCo commands into pilots read-backs, e.g., "descend to" → "descending to".
3. Gather and annotate at least 1 hour of ATCo speech and 1k samples for training your ***high-level entity parser*** system. If the reader is interested in obtaining a general idea of how much data are needed for reaching a desired WER or F1-score, see [62] for ASR and [11] for ATC-NLU.
4. Fine-tune a strong pre-trained ASR model, e.g., Wav2Vec 2.0 or XLSR [26,27] with the ATC audio recordings. For instance, if the performance is not sufficient, you can use open-source corpora (see Table 2) to increase the amount of annotated samples (see [11,30–32]). We recommend acquiring the ATCO2-PL dataset [11], which has

proven to be a good starting point when no in-domain data are available. This is related to ASR and NLU for ATC.

5. Fine-tune a strong pre-trained NLP model, e.g., BERT [28] or RoBERTa [67], with the NLP tags. If the performance is not sufficient, one can follow several text-based data-augmentation techniques. For example, it is possible to replace the callsign in one training sample with different ones from a predefined callsign list. In that way, one can generate many more valuable training samples. It is also possible to use more annotations during fine-tuning, e.g., see the `ATCO2-4h` corpus [11].
6. Lastly, in case you need to adapt the TTS module to pilot speech, you could adapt the FastSpeech2 [29] system. Then, you need to invert the annotations used for ASR, i.e., using the transcripts as input and the ATCo or pilot recordings as targets. This step is not strictly necessary, as already-available pre-trained modules possess a good quality.

7. Conclusions

In this paper, we have presented a novel virtual simulation-pilot system designed for ATCo training. Our system utilizes cutting-edge open-source ASR, NLP and TTS systems. To the best of our knowledge, this is the first such system that relies on open-source ATC resources. The virtual simulation-pilot system is developed for ATCo training purposes; thus, this work represents an important contribution to the field of aviation training.

Our system employs a multi-stage approach, including ASR transcription, a ***high-level entity parser*** system and a repetition-generator module to provide pilot-like responses to ATC communications. By utilizing open-source AI models and public databases, we have developed a simple and efficient system that can be easily replicated and adapted for different training scenarios. For instance, we tested our ASR system on different well-known ATC-related projects, i.e., HAAWAII, MALORCA and ATCO2. We reached as low as 5.5% WER on high-quality data (MALORCA, ATCo speech in operations room) and 15.9% WER on low-quality ATC audio such as the test sets from the ATCO2 project (noise levels below 15 dB).

Going forward, there is significant potential for further improvements and expansions to the proposed system. Incorporating contextual data, such as runway numbers or waypoint lists, could enhance the accuracy and effectiveness of the ASR and high-level entity parser modules. In this work, we evaluated the introduction of real-time surveillance data, which proved to further improve the system's performance in recognizing and responding to ATC communications. For instance, our boosting technique brings a 9% absolute amelioration in callsign-detection accuracy levels (86.7% $\rightarrow$ 96.1%) for the NATS test set. It is also important to recall that additional efforts could be made to fine-tune the TTS system for the improved synthesis of male or female voices, as well as modifying the speech rate, noise artifacts and other features.

The proposed ASR system can reach as low as 5.5% and 15.9% word error rates (WERs) on high- and low-quality ATC audio (Vienna and ATCO2-test-set-1h, respectively). It is also proven that adding surveillance data to the ASR can yield a callsign detection accuracy of more than 96%. Overall, this work represents a promising first step towards developing advanced virtual simulation-pilot systems for ATCo training, and it is expected that future work will continue to explore this research direction.

Author Contributions: Conceptualization, J.Z.-G., A.P., I.N. and P.M.; Data curation, J.Z.-G., A.P. and I.N.; Formal analysis, J.Z.-G., I.N. and M.K.; Funding acquisition, P.M.; Investigation, J.Z.-G., A.P. and I.N.; Methodology, J.Z.-G., A.P., I.N., P.M. and M.K.; Project administration, P.M.; Resources, J.Z.-G. and A.P.; Software, J.Z.-G., A.P., I.N. and P.M.; Supervision, P.M.; Validation, J.Z.-G. and I.N.; Visualization, J.Z.-G.; Writing—original draft, J.Z.-G., I.N. and M.K.; Writing—review and editing, J.Z.-G., P.M. and M.K. All authors have read and agreed to the published version of the manuscript.

Funding: Ideas presented in paper are partially related to needs on simulator pilot operator services described by Eurocontrol in 2020: call for tenders No.: 20-220319-A, title: Simulator Pilot Operator in support to Real-Time Simulations at: https://www.eurocontrol.int/sites/default/files/2020-08/20-220319-a.pdf, accessed on 12 May 2023. This work was also partly supported by SESAR Joint Undertaking under Grant Agreement No. 884287—HAAWAII (**H**ighly **A**utomated **A**ir traffic controller **W**orkstations with **A**rtificial **I**ntelligence **I**ntegration).

Data Availability Statement: Private and public databases are used in this paper. They are covered in detail in Table 2.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ASR	Automatic Speech Recognition
NLP	Natural Language Processing
ATCo	Air Traffic Controller
ATC	Air Traffic Control
CPDLC	Controller-Pilot Data Link Communications
AI	Artificial Inteligencce
WER	Word Error Rate
ML	Machine Learning
VAD	Voice Activity Detection
ATM	Air Traffic Management
TTS	Text-To-Speech
NLU	Natural Language Understanding
PTT	Push-To-Talk
LM	Language Model
AM	Acoustic Model
WFST	Weighted Finite State Transducer
FST	Finite State Transducer
HMM	Hidden Markov Models
DNN	Deep Neural Networks
MFCCs	Mel-frequency Cepstral Coefficients
LF-MMI	Lattice-Free Maximum Mutual Information
VHF	Very-High Frequency
E2E	End-To-End
CTC	Connectionist Temporal Classification
NER	Named Entity Recognition
RG	Response Generator
ICAO	International Civil Aviation Organization
SNR	Signal-To-Noise
dB	Decibel
RBE	Read-back Error
ATCC	Air Traffic Control Corpus
ELDA	European Language Resources Association
ANSPs	Air Navigation Service Providers
CNN	Convolutional Neural Network
GELU	Gaussian Error Linear Units
Conformer	Convolution-augmented Transformer

References

1. Nassif, A.B.; Shahin, I.; Attili, I.; Azzeh, M.; Shaalan, K. Speech recognition using deep neural networks: A systematic review. *IEEE Access* **2019**, *7*, 19143–19165. [CrossRef]
2. Otter, D.W.; Medina, J.R.; Kalita, J.K. A survey of the usages of deep learning for natural language processing. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 604–624. [CrossRef] [PubMed]
3. Zhang, L.; Wang, S.; Liu, B. Deep learning for sentiment analysis: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1253. [CrossRef]

4. Lugosch, L.; Ravanelli, M.; Ignoto, P.; Tomar, V.S.; Bengio, Y. Speech Model Pre-Training for End-to-End Spoken Language Understanding. In Proceedings of the Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15–19 September 2019; pp. 814–818. [CrossRef]
5. Beek, B.; Neuberg, E.; Hodge, D. An assessment of the technology of automatic speech recognition for military applications. *IEEE Trans. Acoust. Speech Signal Process.* **1977**, *25*, 310–322. [CrossRef]
6. Hamel, C.J.; Kotick, D.; Layton, M. *Microcomputer System Integration for Air Control Training*; Technical Report; Naval Training Systems Center: Orlando, FL, USA, 1989.
7. Matrouf, K.; Gauvain, J.; Neel, F.; Mariani, J. Adapting probability-transitions in DP matching processing for an oral task-oriented dialogue. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Albuquerque, NM, USA, 3–6 April 1990; pp. 569–572.
8. Helmke, H.; Ohneiser, O.; Mühlhausen, T.; Wies, M. Reducing controller workload with automatic speech recognition. In Proceedings of the 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), Sacramento, CA, USA, 25–29 September 2016; pp. 1–10.
9. Helmke, H.; Ohneiser, O.; Buxbaum, J.; Kern, C. Increasing ATM efficiency with assistant based speech recognition. In Proceedings of the 13th USA/Europe Air Traffic Management Research and Development Seminar, Seattle, WA, USA, 27–30 June 2017.
10. Nigmatulina, I.; Zuluaga-Gomez, J.; Prasad, A.; Sarfjoo, S.S.; Motlicek, P. A two-step approach to leverage contextual data: Speech recognition in air-traffic communications. In Proceedings of the ICASSP, Singapore, 23–27 May 2022.
11. Zuluaga-Gomez, J.; Veselỳ, K.; Szöke, I.; Motlicek, P.; Kocour, M.; Rigault, M.; Choukri, K.; Prasad, A.; Sarfjoo, S.S.; Nigmatulina, I.; et al. ATCO2 corpus: A Large-Scale Dataset for Research on Automatic Speech Recognition and Natural Language Understanding of Air Traffic Control Communications. *arXiv* **2022**, arXiv:2211.04054.
12. Kocour, M.; Veselý, K.; Blatt, A.; Zuluaga-Gomez, J.; Szöke, I.; Cernocký, J.; Klakow, D.; Motlícek, P. Boosting of Contextual Information in ASR for Air-Traffic Call-Sign Recognition. In Proceedings of the Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August–3 September 2021; pp. 3301–3305. [CrossRef]
13. Zuluaga-Gomez, J.; Sarfjoo, S.S.; Prasad, A.; Nigmatulina, I.; Motlicek, P.; Ondre, K.; Ohneiser, O.; Helmke, H. BERTraffic: BERT-based Joint Speaker Role and Speaker Change Detection for Air Traffic Control Communications. In Proceedings of the 2022 IEEE Spoken Language Technology Workshop (SLT), Doha, Qatar, 9–12 January 2023.
14. Lin, Y.; Li, Q.; Yang, B.; Yan, Z.; Tan, H.; Chen, Z. Improving speech recognition models with small samples for air traffic control systems. *Neurocomputing* **2021**, *445*, 287–297. [CrossRef]
15. Zuluaga-Gomez, J.; Veselỳ, K.; Blatt, A.; Motlicek, P.; Klakow, D.; Tart, A.; Szöke, I.; Prasad, A.; Sarfjoo, S.; Kolčárek, P.; et al. Automatic call sign detection: Matching air surveillance data with air traffic spoken communications. *Proceedings* **2020**, *59*, 14.
16. Fan, P.; Guo, D.; Lin, Y.; Yang, B.; Zhang, J. Speech recognition for air traffic control via feature learning and end-to-end training. *arXiv* **2021**, arXiv:2111.02654.
17. Helmke, H.; Slotty, M.; Poiger, M.; Herrer, D.F.; Ohneiser, O.; Vink, N.; Cerna, A.; Hartikainen, P.; Josefsson, B.; Langr, D.; et al. Ontology for transcription of ATC speech commands of SESAR 2020 solution PJ. 16-04. In Proceedings of the IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), London, UK, 23–27 September 2018; pp. 1–10.
18. Guo, D.; Zhang, Z.; Yang, B.; Zhang, J.; Lin, Y. Boosting Low-Resource Speech Recognition in Air Traffic Communication via Pretrained Feature Aggregation and Multi-Task Learning. In *IEEE Transactions on Circuits and Systems II: Express Briefs*; IEEE: Piscataway, NJ, USA, 2023.
19. Fan, P.; Guo, D.; Zhang, J.; Yang, B.; Lin, Y. Enhancing multilingual speech recognition in air traffic control by sentence-level language identification. *arXiv* **2023**, arXiv:2305.00170 .
20. Guo, D.; Zhang, Z.; Fan, P.; Zhang, J.; Yang, B. A context-aware language model to improve the speech recognition in air traffic control. *Aerospace* **2021**, *8*, 348. [CrossRef]
21. International Civil Aviation Organization. *ICAO Phraseology Reference Guide*; ICAO: Montreal, QC, Canada, 2020.
22. Bouchal, A.; Had, P.; Bouchaudon, P. The Design and Implementation of Upgraded ESCAPE Light ATC Simulator Platform at the CTU in Prague. In Proceedings of the 2022 New Trends in Civil Aviation (NTCA), Prague, Czech Republic, 26–27 October 2022; pp. 103–108.
23. Lin, Y. Spoken instruction understanding in air traffic control: Challenge, technique, and application. *Aerospace* **2021**, *8*, 65. [CrossRef]
24. Lin, Y.; Wu, Y.; Guo, D.; Zhang, P.; Yin, C.; Yang, B.; Zhang, J. A deep learning framework of autonomous pilot agent for air traffic controller training. *IEEE Trans. Hum.-Mach. Syst.* **2021**, *51*, 442–450. [CrossRef]
25. Prasad, A.; Zuluaga-Gomez, J.; Motlicek, P.; Sarfjoo, S.; Nigmatulina, I.; Vesely, K. Speech and Natural Language Processing Technologies for Pseudo-Pilot Simulator. In Proceedings of the 12th SESAR Innovation Days, Budapest, Hungary, 5–8 December 2022.
26. Baevski, A.; Zhou, Y.; Mohamed, A.; Auli, M. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In Proceedings of the Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, Virtual, 6–12 December 2020.
27. Conneau, A.; Baevski, A.; Collobert, R.; Mohamed, A; Auli, M. Unsupervised cross-lingual representation learning for speech recognition. *arXiv* **2021**, arXiv:2006.13979.

28. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 4171–4186. [CrossRef]
29. Ren, Y.; Hu, C.; Tan, X.; Qin, T.; Zhao, S.; Zhao, Z.; Liu, T. FastSpeech 2: Fast and High-Quality End-to-End Text to Speech. In Proceedings of the 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, 3–7 May 2021.
30. Godfrey, J. *The Air Traffic Control Corpus (ATC0)—LDC94S14A*; Linguistic Data Consortium: Philadelphia, PA, USA, 1994. .
31. Šmídl, L.; Švec, J.; Tihelka, D.; Matoušek, J.; Romportl, J.; Ircing, P. Air traffic control communication (ATCC) speech corpora and their use for ASR and TTS development. *Lang. Resour. Eval.* **2019**, *53*, 449–464. [CrossRef]
32. Hofbauer, K.; Petrik, S.; Hering, H. The ATCOSIM Corpus of Non-Prompted Clean Air Traffic Control Speech. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*; European Language Resources Association (ELRA): Marrakech, Morocco, 2008.
33. Pavlinović, M.; Juričić, B.; Antulov-Fantulin, B. Air traffic controllers' practical part of basic training on computer based simulation device. In Proceedings of the International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 22–26 May 2017; pp. 920–925.
34. Juričić, B.; Varešak, I.; Božić, D. The role of the simulation devices in air traffic controller training. In Proceedings of the International Symposium on Electronics in Traffic, ISEP 2011 Proceedings, Berlin, Germany, 26–28 September 2011.
35. Chhaya, B.; Jafer, S.; Coyne, W.B.; Thigpen, N.C.; Durak, U. Enhancing scenario-centric air traffic control training. In Proceedings of the 2018 AIAA modeling and Simulation Technologies Conference, Kissimmee, FL, USA, 8–12 January 2018; p. 1399.
36. Updegrove, J.A.; Jafer, S. Optimization of air traffic control training at the Federal Aviation Administration Academy. *Aerospace* **2017**, *4*, 50. [CrossRef]
37. Eide, A.W.; Ødegård, S.S.; Karahasanović, A. A post-simulation assessment tool for training of air traffic controllers. In *Human Interface and the Management of Information. Information and Knowledge Design and Evaluation*; Springer: Cham, Switzerland, 2014; pp. 34–43.
38. Némethová, H.; Bálint, J.; Vagner, J. The education and training methodology of the air traffic controllers in training. In Proceedings of the International Conference on Emerging eLearning Technologies and Applications (ICETA), Starý Smokovec, Slovakia, 21–22 November 2019; pp. 556–563.
39. Zhang, J.; Zhang, P.; Guo, D.; Zhou, Y.; Wu, Y.; Yang, B.; Lin, Y. Automatic repetition instruction generation for air traffic control training using multi-task learning with an improved copy network. *Knowl.-Based Syst.* **2022**, *241*, 108232. [CrossRef]
40. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
41. Mohri, M.; Pereira, F.; Riley, M. Weighted finite-state transducers in speech recognition. *Comput. Speech Lang.* **2002**, *16*, 69–88. [CrossRef]
42. Mohri, M.; Pereira, F.; Riley, M. Speech recognition with weighted finite-state transducers. In *Springer Handbook of Speech Processing*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 559–584.
43. Riley, M.; Allauzen, C.; Jansche, M. 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*; Association for Computational Linguistics: Boulder, CO, USA, 2009; pp. 9–10.
44. Veselý, K.; Ghoshal, A.; Burget, L.; Povey, D. Sequence-discriminative training of deep neural networks. *Interspeech* **2013**, *2013*, 2345–2349.
45. Bourlard, H.A.; Morgan, N. *Connectionist Speech Recognition: A Hybrid Approach*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1993; Volume 247.
46. Povey, D.; Cheng, G.; Wang, Y.; Li, K.; Xu, H.; Yarmohammadi, M.; Khudanpur, S. Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks. In Proceedings of the Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2–6 September 2018; pp. 3743–3747. [CrossRef]
47. Zuluaga-Gomez, J.; Motlicek, P.; Zhan, Q.; Veselỳ, K.; Braun, R. Automatic Speech Recognition Benchmark for Air-Traffic Communications. *Proc. Interspeech* **2020**, 2297–2301. [CrossRef]
48. Srinivasamurthy, A.; Motlícek, P.; Himawan, I.; Szaszák, G.; Oualil, Y.; Helmke, H. Semi-Supervised Learning with Semantic Knowledge Extraction for Improved Speech Recognition in Air Traffic Control. In Proceedings of the Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, 20–24 August 2017; pp. 2406–2410.
49. Zuluaga-Gomez, J.; Nigmatulina, I.; Prasad, A.; Motlicek, P.; Veselỳ, K.; Kocour, M.; Szöke, I. Contextual Semi-Supervised Learning: An Approach to Leverage Air-Surveillance and Untranscribed ATC Data in ASR Systems. *Proc. Interspeech* **2021**, 3296–3300. [CrossRef]
50. Kocour, M.; Veselý, K.; Szöke, I.; Kesiraju, S.; Zuluaga-Gomez, J.; Blatt, A.; Prasad, A.; Nigmatulina, I.; Motlíček, P.; Klakow, D.; et al. Automatic processing pipeline for collecting and annotating air-traffic voice communication data. *Eng. Proc.* **2021**, *13*, 8.
51. Chen, S.; Kopald, H.; Avjian, B.; Fronzak, M. Automatic Pilot Report Extraction from Radio Communications. In Proceedings of the 2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC), Portsmouth, VA, USA, 18–22 September 2022; pp. 1–8.

52. Povey, D.; Ghoshal, A.; Boulianne, G.; Burget, L.; Glembek, O.; Goel, N.; Hannemann, M.; Motlicek, P.; Qian, Y.; Schwarz, P.; et al. The Kaldi speech recognition toolkit. In Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding, Waikoloa, HI, USA, 11–15 December 2011.
53. Madikeri, S.; Tong, S.; Zuluaga-Gomez, J.; Vyas, A.; Motlicek, P.; Bourlard, H. Pkwrap: A pytorch package for lf-mmi training of acoustic models. *arXiv* **2020**, arXiv:2010.03466.
54. Graves, A.; Jaitly, N. 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; Volume 32, pp. 1764–1772.
55. Chorowski, J.; Bahdanau, D.; Serdyuk, D.; Cho, K.; Bengio, Y. Attention-Based Models for Speech Recognition. In Proceedings of the Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, Montreal, QC, Canada, 7–12 December 2015; pp. 577–585.
56. Schneider, S.; Baevski, A.; Collobert, R.; Auli, M. wav2vec: Unsupervised Pre-Training for Speech Recognition. In Proceedings of the Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15–19 September 2019; pp. 3465–3469. [CrossRef]
57. Baevski, A.; Mohamed, A. Effectiveness of Self-Supervised Pre-Training for ASR. In Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, 4–8 May 2020; pp. 7694–7698. [CrossRef]
58. Zhang, Z.Q.; Song, Y.; Wu, M.H.; Fang, X.; Dai, L.R. Xlst: Cross-lingual self-training to learn multilingual representation for low resource speech recognition. *arXiv* **2021**, arXiv:2103.08207.
59. Chen, S.; Wang, C.; Chen, Z.; Wu, Y.; Liu, S.; Chen, Z.; Li, J.; Kanda, N.; Yoshioka, T.; Xiao, X.; et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *arXiv* **2021**, arXiv:2110.13900.
60. Oord, A.v.d.; Li, Y.; Vinyals, O. Representation learning with contrastive predictive coding. *arXiv* **2018**, arXiv:1807.03748.
61. Baevski, A.; Schneider, S.; Auli, M. vq-wav2vec: Self-Supervised Learning of Discrete Speech Representations. In Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020.
62. Zuluaga-Gomez, J.; Prasad, A.; Nigmatulina, I.; Sarfjoo, S.; Motlicek, P.; Kleinert, M.; Helmke, H.; Ohneiser, O.; Zhan, Q. How Does Pre-trained Wav2Vec2.0 Perform on Domain Shifted ASR? An Extensive Benchmark on Air Traffic Control Communications. In Proceedings of the IEEE Spoken Language Technology Workshop (SLT), Doha, Qatar, 9–12 January 2023.
63. Yadav, V.; Bethard, S. 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, NM, USA, 20–26 August 2018; Association for Computational Linguistics: Santa Fe, NM, USA, 2018; pp. 2145–2158.
64. Grishman, R.; Sundheim, B. Message Understanding Conference- 6: A Brief History. In Proceedings of the COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics, Copenhagen, Denmark, 5–9 August 1996 .
65. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
66. Piskorski, J.; Pivovarova, L.; Šnajder, J.; Steinberger, J.; Yangarber, R. 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*; Association for Computational Linguistics: Valencia, Spain, 2017; pp. 76–85. [CrossRef]
67. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
68. He, P.; Liu, X.; Gao, J.; Chen, W. Deberta: Decoding-Enhanced Bert with Disentangled Attention. In Proceedings of the 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, 3–7 May 2021.
69. Klatt, D.H. Review of text-to-speech conversion for English. *J. Acoust. Soc. Am.* **1987**, *82*, 737–793. [CrossRef]
70. Murray, I.R.; Arnott, J.L.; Rohwer, E.A. Emotional stress in synthetic speech: Progress and future directions. *Speech Commun.* **1996**, *20*, 85–91. [CrossRef]
71. Tokuda, K.; Nankaku, Y.; Toda, T.; Zen, H.; Yamagishi, J.; Oura, K. Speech synthesis based on hidden Markov models. *Proc. IEEE* **2013**, *101*, 1234–1252. [CrossRef]
72. Wang, Y.; Skerry-Ryan, R.J.; Stanton, D.; Wu, Y.; Weiss, R.J.; Jaitly, N.; Yang, Z.; Xiao, Y.; Chen, Z.; Bengio, S.; et al. Tacotron: Towards End-to-End Speech Synthesis. In Proceedings of the Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, 20–24 August 2017; pp. 4006–4010.
73. Shen, J.; Pang, R.; Weiss, R.J.; Schuster, M.; Jaitly, N.; Yang, Z.; Chen, Z.; Zhang, Y.; Wang, Y.; Ryan, R.; et al. Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, 15–20 April 2018; pp. 4779–4783. [CrossRef]
74. Kaur, N.; Singh, P. Conventional and contemporary approaches used in text to speech synthesis: A review. *Artif. Intell. Rev.* **2022**, 1–44. [CrossRef]
75. Jeong, M.; Kim, H.; Cheon, S.J.; Choi, B.J.; Kim, N.S. Diff-TTS: A Denoising Diffusion Model for Text-to-Speech. In Proceedings of the Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August–3 September 2021; pp. 3605–3609. [CrossRef]
76. Sarfjoo, S.S.; Madikeri, S.R.; Motlícek, P. Speech Activity Detection Based on Multilingual Speech Recognition System. In Proceedings of the Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August–3 September 2021; pp. 4369–4373. [CrossRef]

77. Ariav, I.; Cohen, I. An end-to-end multimodal voice activity detection using wavenet encoder and residual networks. *IEEE J. Sel. Top. Signal Process.* **2019**, *13*, 265–274. [CrossRef]
78. Ding, S.; Wang, Q.; Chang, S.y.; Wan, L.; Moreno, I.L. Personal VAD: Speaker-conditioned voice activity detection. *arXiv* **2019**, arXiv:1908.04284.
79. Medennikov, I.; Korenevsky, M.; Prisyach, T.; Khokhlov, Y.Y.; Korenevskaya, M.; Sorokin, I.; Timofeeva, T.; Mitrofanov, A.; Andrusenko, A.; Podluzhny, I.; et al. Target-Speaker Voice Activity Detection: A Novel Approach for Multi-Speaker Diarization in a Dinner Party Scenario. In Proceedings of the Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25–29 October 2020; pp. 274–278. [CrossRef]
80. Ng, T.; Zhang, B.; Nguyen, L.; Matsoukas, S.; Zhou, X.; Mesgarani, N.; Veselỳ, K.; Matějka, P. Developing a speech activity detection system for the DARPA RATS program. In Proceedings of the Thirteenth Annual Conference of the International Speech Communication Association, Portland, OR, USA, 9–13 September 2012.
81. Helmke, H.; Ondřej, K.; Shetty, S.; Arilíusson, H.; Simiganoschi, T.; Kleinert, M.; Ohneiser, O.; Ehr, H.; Zuluaga-Gomez, J.; Smrz, P. Readback Error Detection by Automatic Speech Recognition and Understanding—Results of HAAWAII Project for Isavia's Enroute Airspace. In Proceedings of the 12th SESAR Innovation Days, Budapest , Hungary, 5–8 December 2022.
82. Cordero, J.M.; Dorado, M.; de Pablo, J.M. Automated speech recognition in ATC environment. In Proceedings of the 2nd International Conference on Application and Theory of Automation in Command and Control Systems, London, UK, 29–31 May 2012; pp. 46–53.
83. Delpech, E.; Laignelet, M.; Pimm, C.; Raynal, C.; Trzos, M.; Arnold, A.; Pronto, D. A Real-life, French-accented Corpus of Air Traffic Control Communications. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*; European Language Resources Association (ELRA): Miyazaki, Japan, 2018.
84. Segura, J.; Ehrette, T.; Potamianos, A.; Fohr, D.; Illina, I.; Breton, P.; Clot, V.; Gemello, R.; Matassoni, M.; Maragos, P. The HIWIRE Database, a Noisy and Non-Native English Speech Corpus for Cockpit Communication. 2007. Available online: http://www.hiwire.org (accessed on 12 May 2023).
85. Gulati, A.; Qin, J.; Chiu, C.; Parmar, N.; Zhang, Y.; Yu, J.; Han, W.; Wang, S.; Zhang, Z.; Wu, Y.; et al. Conformer: Convolution-augmented Transformer for Speech Recognition. In Proceedings of the Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25–29 October 2020; pp. 5036–5040. [CrossRef]
86. LeCun, Y.; Bengio, Y. Convolutional networks for images, speech, and time series. *Handb. Brain Theory Neural Netw.* **1995**, *3361*, 1995.
87. Vyas, A.; Madikeri, S.; Bourlard, H. Lattice-Free Mmi Adaptation of Self-Supervised Pretrained Acoustic Models. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 6219–6223. [CrossRef]
88. Ravanelli, M.; Parcollet, T.; Plantinga, P.; Rouhe, A.; Cornell, S.; Lugosch, L.; Subakan, C.; Dawalatabad, N.; Heba, A.; Zhong, J.; et al. SpeechBrain: A general-purpose speech toolkit. *arXiv* **2021**, arXiv:2106.04624.
89. Panayotov, V.; Chen, G.; Povey, D.; Khudanpur, S. Librispeech: An ASR corpus based on public domain audio books. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Australia, 19–24 April 2015; pp. 5206–5210. [CrossRef]
90. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
91. Hendrycks, D.; Gimpel, K. Gaussian error linear units (gelus). *arXiv* **2016**, arXiv:1606.08415.
92. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
93. Karita, S.; Chen, N.; Hayashi, T.; Hori, T.; Inaguma, H.; Jiang, Z.; Someki, M.; Soplin, N.E.Y.; Yamamoto, R.; Wang, X.; et al. A comparative study on transformer vs rnn in speech applications. In Proceedings of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Singapore, 14–18 December 2019; pp. 449–456.
94. Graves, A.; Graves, A. Connectionist temporal classification. In *Supervised Sequence Labelling with Recurrent Neural Networks*; Springer: Berlin/Heidelberg, Switzerland, 2012; pp. 61–93.
95. Chen, Z.; Jain, M.; Wang, Y.; Seltzer, M.L.; Fuegen, C. End-to-end Contextual Speech Recognition Using Class Language Models and a Token Passing Decoder. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, UK, 12–17 May; pp. 6186–6190. [CrossRef]
96. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. Transformers: State-of-the-Art Natural Language Processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, 16–20 November 2020; pp. 38–45. [CrossRef]
97. Lhoest, Q.; Villanova del Moral, A.; Jernite, Y.; Thakur, A.; von Platen, P.; Patil, S.; Chaumond, J.; Drame, M.; Plu, J.; Tunstall, L.; et al. Datasets: A Community Library for Natural Language Processing. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, 7–11 November 2021; pp. 175–184. [CrossRef]

98. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16–21 June 2013; Volume 28, pp. 1310–1318.
99. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. In Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019.