Journal of Open Aviation Science (2026), Vol.**4**
doi:10.59490/joas.2026.8477

JOAS

PROCEEDINGS | *The 13th OpenSky Symposium*

Can YouTube Stream Recordings Improve Speech Recognition for Air Traffic Control?

Niclas Wüstenbecker [*], Oliver Ohneiser [1] and Matthias Kleinert [1]

German Aerospace Center (DLR), Institute of Flight Guidance, Department Controller Assistance Systems, Lilienthalplatz 7, 38108 Braunschweig, Germany

*Corresponding author: niclas.wuestenbecker@dlr.de

(Received: 3 Nov 2025; Revised: 10 Mar 2026; Accepted: 18 Mar 2026; Published: 20 Mar 2026)
(Editor: Junzi Sun; Reviewers: Allan Tart and Max Li)

Abstract

Automatic speech recognition for air traffic control (ATC) faces severe training data scarcity due to operational recording restrictions and expensive domain-expert transcription requirements. We address this limitation by developing an automated pipeline that extracts large-scale, high-quality training data from publicly available YouTube streams of virtual ATC simulator sessions from networks such as VATSIM and IVAO. Our approach systematically processes over 2,000 hours of content spanning 709 videos from virtual airports and airspaces in 17 countries across multiple continents, operational domains (ground, tower, approach, en-route), and diverse speaker accents. The pipeline employs speaker diarization for utterance segmentation, parallel transcription using three complementary automatic speech recognition (ASR) architectures with distinct error characteristics, and Large Language Model-based transcript fusion that synthesizes improved pseudo-labels while filtering non-ATC content. Manual verification on a stratified 120-minute evaluation set demonstrates 10.2% word error rate for controller speech and 18.3% for pilot speech—representing 37% relative improvement over the best individual model and establishing pseudo-label quality sufficient for downstream model training. We show the feasibility of this approach by training a compact 115M-parameter ASR model exclusively on automatically generated transcripts without any manually annotated operational data. Evaluation on the operational ATCO2 benchmark reveals 21.1% word error rate compared to 35.6% for published baselines trained on smaller manually-transcribed datasets, despite the domain gap between virtual and operational ATC, while achieving approximately five times faster inference. These results demonstrate that large-scale geographically and acoustically diverse, pseudo-labeled data can effectively compensate for moderate label noise when training specialized-domain speech recognition systems. We openly release the complete processing pipeline, curated video collection, and our trained model to enable reproducible research.

Keywords: Air Traffic Control; Automatic Speech Recognition; Public Dataset; Large Language Model;

Abbreviations: ATC: Air Traffic Control, ASR: Automatic Speech Recognition, WER: Word Error Rate, ATCO: Air Traffic Controller, LLM: Large Language Model, VATSIM: Virtual Air Traffic SIMulation Network, IVAO: International Virtual Aviation Organisation,

1. Introduction

Automatic Speech Recognition (ASR) systems are increasingly critical for modern applications, yet general-purpose models like Whisper [1] remain unsuitable for use in specialized domains such as Air Traffic Control (ATC) where the communication patterns diverge substantially from standard

English. ATC communications present multiple unique challenges: controllers and pilots communicate using the standardized ICAO phonetic alphabet and specialized phraseology, with faster speech rates averaging 6 words per second versus 4 in natural conversation [2]. Moreover, ATC communications employ domain-specific terminology such as waypoint identifiers, airline callsigns, and airport codes, which are critical for ensuring operational safety but occur very rarely in general-domain training corpora.

Progress in ATC ASR has been severely constrained by data scarcity. Existing public datasets typically contain only a few hours of audio, are geographically limited to single airports or control centers, and exhibit minimal speaker diversity and accent variation. They rarely span multiple operational domains (ground, tower, approach, en-route) and are thus less robust in slightly different recording conditions. Manually creating large-scale ATC datasets faces substantial barriers: access to an operational environment remains limited to very few entities, the transcription demands expensive domain expertise, and operational recordings contain sensitive information subject to legal restrictions. For these reasons, most existing datasets prioritize data quality over scale.

We address this gap by exploiting publicly available YouTube stream recordings from virtual ATC simulator networks, where virtual controllers practice realistic ATC procedures. While simulator phraseology may deviate from operational standards, these streams provide enormous scale as thousands of archived hours featuring diverse speakers, regions, accents, and operational domains. We hypothesize that this data contains a substantial vocabulary and communication pattern overlap with real-world operations, which can be used to improve ASR for real ATC communications.

Our proposed pipeline can be used for automated data extraction of ATC-relevant speech from a collection of YouTube videos. We download and extract the audio data, perform speaker diarization and segment audio into short 1–20 second utterances. Then, we combine the outputs of multiple ASR models with utterance transcription fusion using a Large Language Model (LLM) to exploit complementary error characteristics. Furthermore, the LLM is used to differentiate between relevant ATC-like content and general commentary. We perform a manual evaluation of a 120-minute subset spanning multiple speakers, regions, and accents to assess the quality of the automatic transcriptions. To facilitate reproducible research, we publish our processing pipeline, the used video collection, and trained model weights.

To summarize, our contributions are:

- a large-scale, diverse ATC-style dataset collected from YouTube content spanning multiple speakers, accents, regions, and operational domains,
- an automated, extendable dataset creation pipeline using an LLM for transcript fusion and content filtering,
- an evaluation of the LLM transcript fusion quality, a comparison with other ASR model performances, an assessment of how beneficial this additional dataset is for the purpose of ATC ASR, and
- the open-source release of our code[1] and trained model weights[2] to enable reproducing, extending, or transferring the method to new domains.

The remainder of this paper is organized as follows. Section 2 reviews existing ATC datasets, ASR architectures, and transcript fusion methods. Section 3 describes our automated pipeline including video collection, audio preprocessing, speaker diarization, and LLM-based transcript fusion. Section 4 evaluates dataset diversity and transcription quality through systematic manual verification.

[1]The code is published openly at: https://github.com/niclaswue/youtube-atc
[2]The model is available at: https://huggingface.co/niclaswue/youtube-atc-fastconformer

Section 5 trains a distilled ASR model on the automatically generated transcripts and evaluates performance on operational benchmarks. Section 6 summarizes findings and future research directions.

2. Related Work

2.1 Data Resources

The development of automatic speech recognition for ATC has been constrained by limited training data. While high-quality datasets exist, including the UWB-ATCC corpus (20 hours from Czech airspace) [3], the LDC-ATCC corpus (26 hours from US airports) [4], and the ATCOSIM corpus (10 hours of simulated communications) [5], their scale remains insufficient for training robust modern ASR systems. The ATCO2 dataset [6] addresses this limitation through pseudo-labeling, providing over 5,000 hours of automatically transcribed real ATC communications. However, the pseudo-labeled transcripts exhibit approximately 30% WER due to the challenging acoustic conditions of real-world Very High Frequency (VHF) radio recordings, making the data difficult to use effectively for downstream applications. Furthermore, ATCO2 is not publicly available, with only a 1-hour test set freely accessible for research, while the full dataset requires purchase. Inspired by ATCO2's pseudo-labeling strategy, our work employs LLM-based transcript fusion to enable a modernized approach that achieves substantially higher label quality. Beyond aviation, the recently published YODAS dataset [7] demonstrates that YouTube provides an excellent source for large-scale ASR training data, as content creators upload diverse, long-form recordings with natural acoustic variability that, despite imperfect quality, effectively support current state-of-the-art model training [8]. We extend this paradigm to ATC by exploiting virtual controller streams that provide similar scale and diversity benefits while offering superior audio quality compared to VHF radio recordings.

2.2 Modelling Techniques

General-purpose ASR models such as Whisper [1], trained on 680,000 hours of multilingual data, demonstrate strong zero-shot capabilities but require domain-specific fine-tuning for specialized applications like ATC, where untrained performance yields WERs exceeding 90%. Recent models like Voxtral Mini Transcribe [9] achieve near state-of-the-art performance with native semantic understanding capabilities. For ATC applications, fine-tuned models show dramatic improvements, with Zuluaga et al. [10] demonstrating that even limited in-domain data yields substantial WER reductions over hybrid baselines when applied to Wav2Vec 2.0 architectures.

Combining outputs from multiple ASR systems has emerged as an effective strategy for improving transcription accuracy. Traditional approaches employ voting algorithms to select among multiple hypotheses [11], with domain-specific fusion mechanisms subsequently developed for ATC applications [12]. Recent advances leverage LLMs for more sophisticated fusion and error correction. Chen et al. [13] propose Uncertainty-Aware Dynamic Fusion, which dynamically integrates acoustic information during decoding, while Hsu et al. [14] present Generative Fusion Decoding, achieving up to 17.7% WER reduction through byte-level likelihood calculation. Prakash et al. [15] introduce a unified framework using LLMs for ensemble output correction, replacing traditional voting with generative correction for improved pseudo-label quality in semi-supervised learning scenarios. Complementing these fusion approaches, post-hoc error correction methods have shown promise: Chen et al. [16] demonstrate that LLMs can correct tokens entirely missing from N-best lists through carefully designed prompts, Li et al. [17] explore multilingual error correction strategies, and Sachdev et al. [18] propose evolutionary prompt optimization for maximum correction performance. To evaluate these approaches beyond traditional metrics, Pulikodan et al. [19] introduce LLM-based quality assessment that accounts for different error types and their operational impact.

3. Methodology: Dataset Creation Pipeline

Our data processing pipeline transforms raw YouTube videos into a curated dataset of individual ATC transmissions with high-quality, single-speaker transcripts. The full dataset creation pipeline, illustrated in Figure 1, consists of the following stages: video collection, audio extraction and preprocessing, audio segmentation, automatic transcription of the audio, and finally LLM-based transcript fusion with content filtering. For each step, we describe the detailed implementation in a separate subsection.

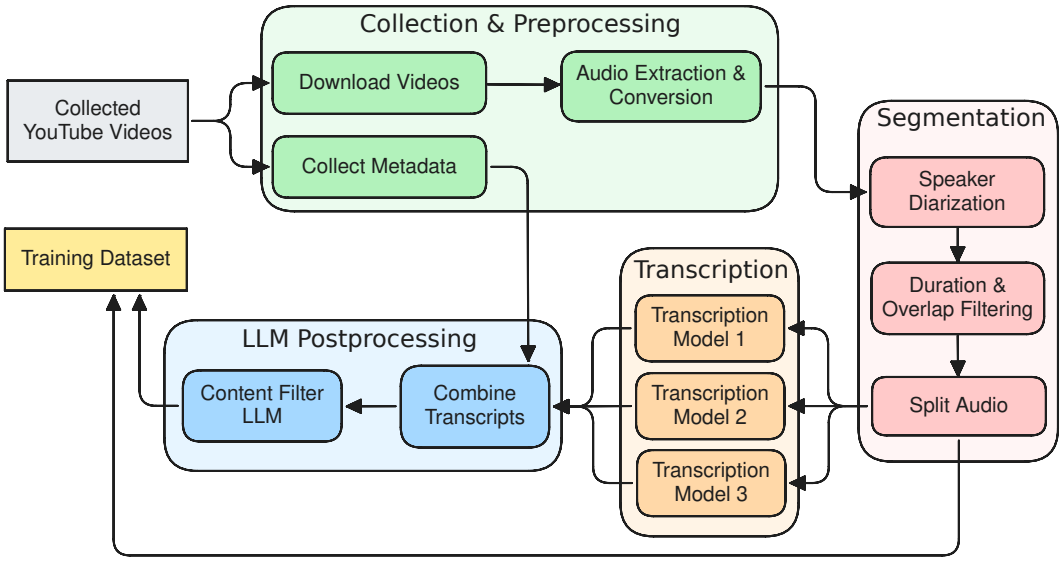

Figure 1. Overview of the Processing Pipeline

3.1 Video Collection Process and Curation Criteria

As a first step, we collect relevant YouTube videos as input data to the pipeline. We collected air traffic control gameplay videos from YouTube, where many enthusiasts upload stream recordings of their virtual ATC sessions. We mainly targeted the virtual ATC platforms VATSIM (Virtual Air Traffic Simulation Network) and IVAO (International Virtual Aviation Organisation) which attract the largest number of players, but also included recordings of other games and platforms, if they were deemed relevant.

The video discovery process employed a manual search strategy using multiple seed search terms such as:

- `VATSIM ATC <placeholder>`
- `IVAO ATC <placeholder>`
- `virtual ATC stream`
- `ATC gameplay`
- `lets play air traffic control`

where placeholders were instantiated with specific countries (e.g., Germany, France, UK) and airports (e.g., London, Frankfurt, Zürich). After a first selection, we used YouTube's recommendation system to identify similar channels and playlists containing long-duration uploads. For each video, a

manual verification of the content was conducted to ensure relevance. During collection, preference was given to stream recordings exceeding 30 minutes that contained substantial ATC communication. We excluded pilot perspective videos, which form the majority of flight simulator content on YouTube, due to the unfavorable ratio of ATC communication to general commentary and moderation which would have to be filtered out. During curation, we also aimed to maximize geographic and speaker diversity to enhance the model robustness across different airports, regions, accents, and phraseology variants.

For each collected video, we extracted metadata including title, description, upload date, duration, and channel information. In addition, we manually assigned a speaker accent to each uploader using the channel information, if possible. A first post-collection analysis of the upload dates, shown in Figure 2, shows that most videos classified as relevant were uploaded in the past two years. This can be attributed to the growing popularity of virtual ATC platforms in many countries such as the United Kingdom [20].

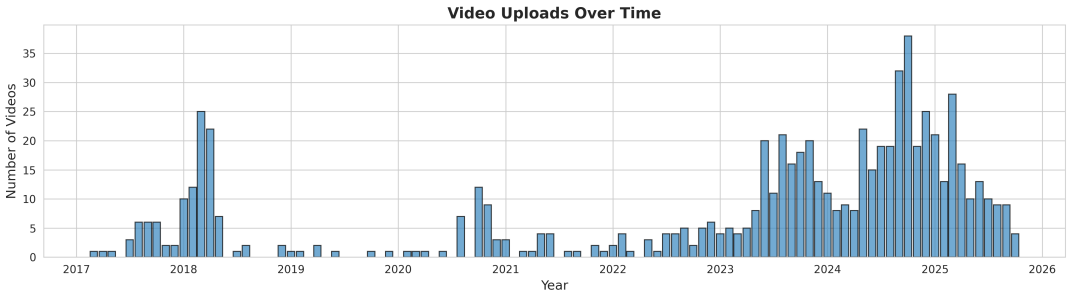

Figure 2. Overview of the video upload dates of all collected videos. Most content classified as relevant was uploaded in the past two years.

The final collection comprises 709 videos totaling more than 2,000 hours of content, requiring approximately 1.5 TB of storage for the extracted, full-length audio files. Table 1 presents the video duration distribution, showing that most videos have a length between two hours and four hours. Around 20% of videos exceed a length of four hours.

Table 1. Distribution of collected videos by duration

Duration Range	Video Count	Percentage (%)
< 15 min	7	1.0
15-30 min	23	3.2
30-60 min	121	17.1
60-120 min	68	9.6
120-240 min	347	48.9
≥ 240 min	143	20.2
Total	709	100.0

During collection, the URLs of all relevant videos were stored line-by-line in a plain text file, which acts as input to the first processing stage of the proposed pipeline.

We deliberately excluded publicly available real-world ATC recordings, cause our paper aims at validating the general setup and workflow of our approach without dealing with the additional challenges of degraded VHF radio audio quality. As this would likely introduce substantially more tran-

scription errors during the multi-model ASR and LLM fusion process. Incorporating such recordings with more aggressive audio preprocessing is a promising direction for future work.

3.2 Audio Extraction and Preprocessing

The list of relevant video URLs created in Section 3.1 is now processed to download and extract the audio data. For each URL, we download and extract the audio using yt-dlp[3]. Each audio file was automatically converted to a 16 kHz sampling rate and mono channel. The 16 kHz standard is widely adopted in ASR model training and is sufficient for speech recognition according to the Nyquist-Shannon sampling theorem. We deliberately avoid applying denoising, audio filters or other audio enhancement techniques to preserve the natural variability and quality differences present in the data. This will ensure the trained model encounters diverse acoustic conditions during training rather than artificially cleaned audio.

3.3 Speaker Diarization and Segmentation

The long audio files obtained in Section 3.2 are not feasible for training ASR models. Therefore, we need to split the audio into shorter utterances, each performed by a single speaker. To achieve this, we employ a Voice Activity Detection (VAD) and speaker diarization pipeline to segment the audio into smaller chunks. We chose the pyannote speaker diarization pipeline [4], which is openly available and widely considered the state-of-the-art open-source solution for this task. During VAD, continuous temporal segments containing speech are identified. The speaker diarization step assigns a speaker ID to each of these segments. Using the speaker IDs, we identify segments with more than one speaker at a time and exclude them from the dataset. We experimented with speaker role classification; however, we found that speaker IDs were not always reliable enough to consistently extract the primary speaker from the videos. Furthermore, we enforce a duration constraint and exclude segments shorter than 1 second and longer than 20 seconds. The minimum threshold filters out brief acknowledgments and many audio artifacts, while the maximum is used to discard long moderation segments or commentary. The distribution of filtered samples after this step is shown in Figure 3. Following the LLM-based classification pipeline, 377 hours were labeled as *irrelevant* and excluded from the training corpus, while the remaining 464 hours distributed across the *ATCO*, *Pilot*, and *unsure* classes constitute the dataset used for downstream model training and evaluation.

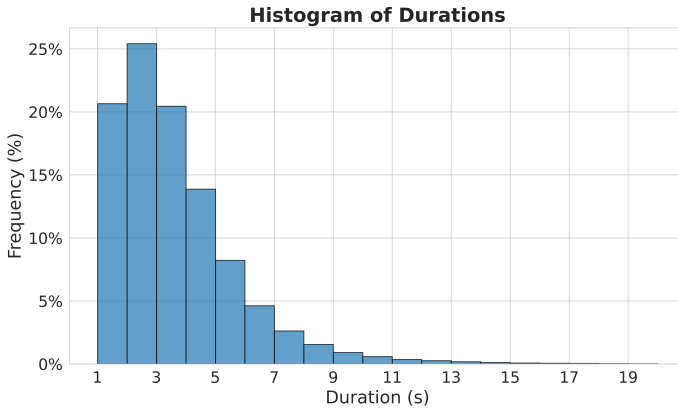

Figure 3. Histogram of sample durations after applying the overlap and duration filter. The vast majority of samples is shorter than 10 seconds.

[3] https://github.com/yt-dlp/yt-dlp
[4] Model available at: https://huggingface.co/pyannote/speaker-diarization-3.1

Finally, these filtered and cleaned segment timestamps are used to split the long audio files into individual short clips which are saved as wave files, ready for further processing. Each resulting clip is treated as an independent utterance throughout all subsequent pipeline stages, including transcription, fusion, and model training. No sequential context or temporal ordering between adjacent segments is preserved. While this design simplifies the pipeline and aligns with standard ASR training practices where utterances are processed in isolation, it also means that discourse-level information, such as the structure of a controller–pilot exchange sequence, is not exploited. Preserving and leveraging such sequential context represents a potential avenue for future improvement.

3.4 Multi-Model Automatic Transcription

To analyze and filter the content of the obtained utterances, we rely on automatically generated transcripts for each utterance. Using a single ASR model, however, is challenging because we cannot be sure if the nature of the content is ATC-specific, requiring a tuned ATC model, or if it is moderation or commentary, best suited for a general-purpose model such as Whisper or Voxtral. Therefore, we decided to implement a transcript fusion approach where we transcribe each audio file with multiple complementary ASR models and use a Large Language Model for combining the results. Due to the large quantity of data, we only fuse the transcripts of selected ASR models, which support fast vLLM inference. This led to the exclusion of Wav2Vec 2.0 models from the fusion process as they are not supported by vLLM. Beside the vLLM requirement the chosen models were selected to have different strengths and complementary error patterns due to their different training and finetuning data. Our first model, OpenAI Whisper v3 Turbo [1], is a large general-domain ASR model trained on internet data. It demonstrates robustness to various accents and audio qualities and handles general vocabulary effectively, but may struggle with ATC-specific terminology. The second model is a Whisper model fine-tuned specifically on existing ATC datasets [21]. This model excels at recognizing aviation-specific terms including callsigns, waypoints, and standard phraseology. However, its specialization may lead to overfitting on ATC domain language, potentially missing or hallucinating general words in streamer commentary that falls outside its training distribution. As a third model, we selected Mistral Voxtral [9], which employs an alternative architecture and training paradigm. While potentially slightly less accurate overall than the Whisper models, it provides valuable model diversity through different error patterns that can be complementary during transcript fusion.

All three models were hosted using a local vLLM server [22] with efficient parallel request processing to maximize model throughput. Each audio segment was transcribed independently by all three models, generating three distinct transcripts per clip. The parallel transcription strategy reveals consensus regions where models agree (indicating high confidence), conflicting transcripts where models disagree (signaling uncertain or ambiguous audio), and model-specific strengths.

The ATC-tuned model more frequently transcribes airline names and aviation terminology correctly compared to general models, and crucially provides verbatim output (spelling out numbers and avoiding punctuation) which the general Whisper and Voxtral models do not, as they include punctuation and use digits in their tokenizer. The general Whisper model typically transcribes streamer commentary and moderator speech more accurately, handles non-standard accents better, but may miss ATC-specific abbreviations. Voxtral provides independent confirmation or alternative hypotheses and handles acoustic ambiguities differently from the Whisper architecture. All transcripts are saved in JSON format and are combined by the LLM in the next processing step.

3.5 LLM-Based Transcript Fusion and Content Filtering

To achieve transcript fusion, we employed Mistral Small 3.2 (fp8 quantization) [23] as our fusion model, a fast LLM with strong reasoning capabilities that fits on a single 48 GB VRAM GPU. We present to the model the different ASR model outputs along with the source video title and descrip-

tion for further guidance during transcription and prompt it to combine the transcripts. As a next step, we use the LLM to classify the combined transcript. To speed up the process and to reduce the computational requirements, we use a single prompt template for transcript combination and transcript classification. The used system prompt is shown in Figure 4.

System Prompt Template for Multi-Model ASR Transcript Fusion and Classification

You are an expert AI system specializing in Air Traffic Control (ATC) communications. Your mission is to analyze multiple ASR transcripts of a single audio utterance, combine them into a single, high-fidelity verbatim transcript, and classify its content.

Inputs
You will receive two pieces of information:

1. **ASR Transcripts**: A JSON object containing transcripts from three different models:
   - **<model_1>**: *<description of model 1 capabilities (i.e. model_1 is trained on aviation data and knows specific terms best) >*
   - **<model_2>**: *<description of model 2 capabilities>*
   - ...
   - **<model_n>**: *<description of model n capabilities>*
2. **Source Context**: A block tagged `<SOURCE_CONTEXT_MAY_HELP>` containing the title and description of the source video. Use this context to help resolve ambiguities, such as airport names or callsigns.

Core Heuristics
You must follow these rules to make your decision. They are listed in order of importance.

1. **<heuristic name 1>**: *<description (i.e. prefer model_1 for aviation terms) >*
2. **<heuristic name 2>**: *<description>*
3. ...
4. **<heuristic name n>**: *<description>*

Procedure
Follow this four-step reasoning process. Perform your reasoning internally before generating the final JSON output.

1. **Analyze & Compare**: Scrutinize all three transcripts. Identify points of agreement and disagreement. Apply the heuristics and use the `<SOURCE_CONTEXT_MAY_HELP>` to inform your analysis.
2. **Synthesize & Combine**: Create the best possible combined transcript based on your analysis. This transcript must adhere to the following rules:
   - Lowercase only.
   - No punctuation.
   - Strictly verbatim: All numbers and initialisms must be spelled out (e.g., “one niner” not “19”, ...).
   - Apply ATC Expertise: Correct obvious ASR errors that violate standard phraseology (e.g., change “ran away” to “runway”).
3. **Classify**: Based on the final combined transcript, assign **one** of the following classifications:
   - `atco`: A clear instruction or clearance from an air traffic controller.
   - `pilot`: A clear readback or request from a pilot.
   - `unsure`: The utterance may contain ATC-related communication, but the speaker’s role is ambiguous
   - `irrelevant`: The utterance is not ATC communication (e.g., background chatter, commentary, ...) or is indecipherable.
4. **Final Output**: Your final response must only be a JSON object enclosed in `<RESULT>` tags. Do not include any of your reasoning notes in the final output.

Examples (Repeated $n_{examples}$ times)
`<INPUT>` {*<transcripts from models>*} `</INPUT>`
`<SOURCE_CONTEXT_MAY_HELP>` *<context>* `</SOURCE_CONTEXT_MAY_HELP>`
`<RESULT>` {*<correct classification, correct combined_transcript>*} `</RESULT>`

Figure 4. Complete prompt template for multi-model ASR transcript fusion in ATC communications. The prompt is filled by inserting transcripts from multiple models with complementary strengths, applies heuristics to synthesize a verbatim transcript, and classifies the utterance type. We use few-shot prompting for in-context learning. The prompt was originally formatted as markdown text.

We engineered the prompt following best practices, including clear instructions on the fusion strategy and in-context examples demonstrating the distinctions between the chosen classes *air traffic controller (ATCO)*, *Pilot*, and *irrelevant speech*. The specified output format as structured JSON, containing both the fused transcript and classification label, was chosen to easily validate the correctness of the response. As mentioned earlier, we also incorporate the video title and description as additional context to potentially help resolve ambiguities such as airport names or airlines. Our main target classes for the dataset were *ATCO* and *Pilot*, which should contain relevant, near-real-world utterances. We included the class *Unsure* for ambiguous cases where it is not clear if the transcript is relevant. Separating the clear from the borderline transcripts helps with increasing quality of the ATCO and Pilot classes. Finally, *Irrelevant* segments contain streamer commentary and moderator speech, such as thanking subscribers, explaining actions to viewers, or off-topic discussion. Filter-

ing these segments significantly reduces dataset noise and focuses the training data on actual ATC communications. To enforce the correct formatting, we include few-shot examples for in-context learning, as suggested in most of the recent literature [24]. To minimize computational overhead associated with the system prompt during inference, we make use of automatic prefix-caching, which enables efficient reuse of previously computed prompt representations.

Our LLM transcript fusion approach leverages language understanding beyond acoustic model capabilities, using contextual knowledge to resolve ambiguities more effectively than simple voting or confidence-based selection methods. This approach also simplifies transforming transcripts into the correct format required for downstream training: verbatim, lowercase text without punctuation. The utterances together with combined transcripts, which are classified as either ATCO or Pilot, now form the basis of our dataset used for further evaluation. The diarization and filtering process systematically removes silence periods and segments with overlapping speech, reducing the corpus from the original 2,000 hours to 841 hours of single-speaker utterances that meet the specified duration and quality constraints.

A key risk in any LLM-based pipeline is hallucination—generating plausible but incorrect transcripts, especially when all upstream ASR models are uncertain. Our approach mitigates this through several safeguards. First, structured JSON output enables automatic validation: non-conforming responses are discarded and reprocessed. Second, the system prompt was iteratively refined on a small held-out subset, containing approximately 100 samples, to ensure generalization. Third, the multi-model input design constrains the LLM to selecting and combining existing ASR hypotheses rather than generating transcripts from scratch, limiting unconstrained hallucination. We validate output quality through word frequency analysis (Section 4.1), confirming that the linguistic distribution matches real-world ATC patterns, and through manual verification on a 120-minute evaluation set (Section 4.2). We acknowledge that additional automated checks—such as ATC grammar validators or multi-LLM consensus mechanisms—could further strengthen confidence and represent a promising future direction.

4. Dataset Evaluation

4.1 Dataset Diversity

To identify potential dataset biases, we evaluate the content of the collected videos across multiple criteria. First, we plot the geographic distribution of airports and air traffic control facilities (i.e., control centers) from which the virtual air traffic controllers were operating. The resulting analysis shown in Figure 5 reveals substantial global coverage, with particularly dense representation in Europe and North America. Additionally, the dataset includes multiple entries from Australia, Africa, and Asia, demonstrating reasonable geographic diversity across continents.

The dataset features many accents, shown in Figure 6a, which is valuable for training a robust ASR model. The distribution of accents exhibits a pronounced imbalance, with German-accented English dominating at 41.47% and European/North American varieties collectively constituting 91.57% of the corpus. This geographic concentration roughly corresponds to the distribution of airports and control centers present in the dataset, suggesting a sampling bias introduced at the data collection stage. The controller working position distribution, shown in Figure 6b, offers more diversity. It was determined by classifying the title and description of the video. When analyzing the distribution of the virtual ATC network or game presented in the videos, we found that more than 88% of videos are related to the *VATSIM* network. The other gameplays include *IVAO*, *ATCpro*, *Tower! Simulator 3*, *Tower!3D Pro* and *London Control ATC*.

Regarding the internal consistency of long-duration videos, the working position classification in

Figure 5. Overview of all virtual airports and virtual control stations collected. Europe and North America are densely covered. Base map geometries generated using Natural Earth data.

Country/Region (ISO)	Videos	%
Germany (DE)	294	41.5%
Ireland (IE)	99	14.0%
United States (US)	86	12.1%
Canada (CA)	77	10.9%
Austria (AT)	30	4.2%
United Kingdom (GB)	28	4.0%
Spain (ES)	27	3.8%
Philippines (PH)	24	3.4%
Jamaica (JM)	19	2.7%
Netherlands (NL)	13	1.8%
India (IN)	4	0.6%
Hong Kong (HK)	3	0.4%
Taiwan (TW)	3	0.4%
Switzerland (CH)	2	0.3%
Total	**709**	**100.0%**

(a) Distribution of English accents as classified via the origin of the uploading channel.

Working Position	Videos	%
Tower	205	28.9%
En-Route	173	24.4%
Approach/Departure	166	23.4%
Ground	102	14.4%
Delivery	9	1.3%
Oceanic	5	0.7%
Unknown	49	6.9%
Total	**709**	**100.0%**

(b) Controller Working Position distribution obtained from classifying the title and description.

Figure 6. Dataset statistics by accent and controller working position.

Figure 6b was derived from each video's title and description metadata. Virtual ATC streams on platforms such as VATSIM typically feature a single controller operating one position per session (e.g., tower or approach), as indicated by the session title. However, occasional position changes during long streams cannot be ruled out entirely. Importantly, since our pipeline segments each video into independent short utterances (1–20 seconds), any within-video domain mixing is mitigated at the individual sample level: each training sample represents a single, self-contained utterance regardless of the broader session context. Potential artifacts inherent to YouTube-sourced content, including intermittent streamer commentary, breaks in operational realism, and background music,

are systematically characterized through the manual quality assessment described in Section 4.2.

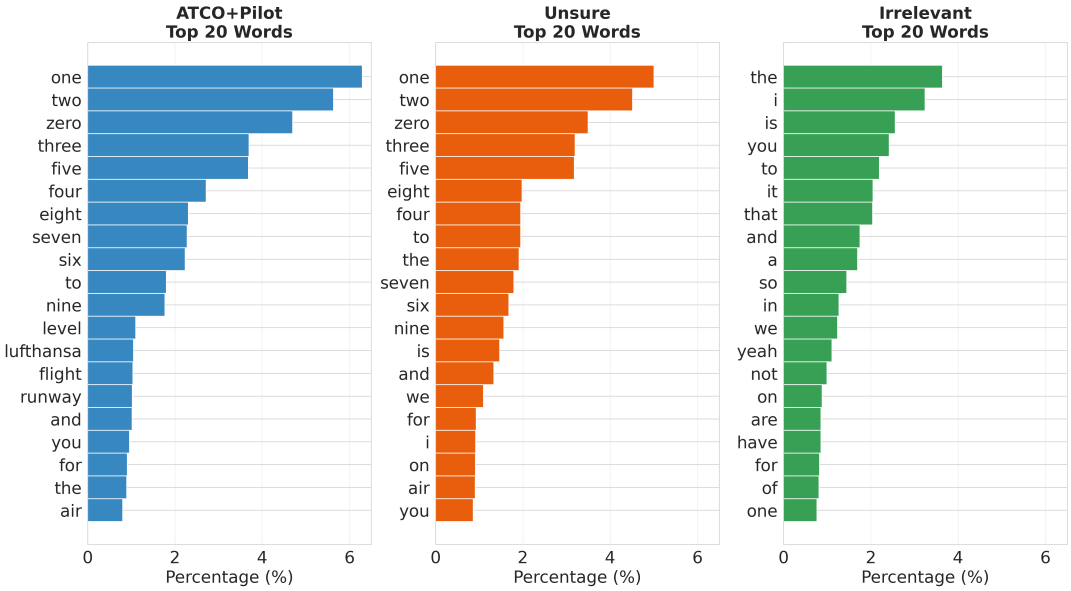

Figure 7. Overview of most frequent words per class. The word frequencies in the ATCO+Pilot classes closely match those found in real-world operational data, while the corpus classified as irrelevant resembles general English word frequencies.

Finally, to evaluate the quality of our LLM-based content filtering approach, we conducted a word frequency analysis of the most common words occurring within each classified category. Figure 7 presents the distribution of high-frequency terms across the different classes. ATCO and pilot utterances demonstrate a pronounced predominance of numerical expressions, likely caused by altitude assignments, heading instructions, and frequency changes. This numerical density closely aligns with the word frequency distributions found by Chen et al. [25] in their analysis of real-world air traffic control communications. In contrast, the irrelevant class exhibits word frequency patterns that correspond closely with general English usage measured in the Corpus of Contemporary American English (COCA) [26], a large English reference corpus. This alignment confirms successful identification and segregation of non-aviation content. The unsure class presents intermediate linguistic characteristics, with its most frequent terms comprising a mixture of standard English vocabulary and domain-specific aviation terminology. This hybrid distribution is consistent with the class's intended function as a boundary category for utterances with ambiguous or uncertain classification confidence.

4.2 Manual Transcription and Ground Truth Dataset

To assess the quality of our automated transcription pipeline, we implemented a systematic multi-stage verification process. First, we curated a representative evaluation dataset comprising 1,708 speech samples with a total duration of 120 minutes. The sample selection strategy ensured balanced representation across accent categories, with each accent group contributing approximately equal speech duration. We restricted our analysis to utterances classified by the LLM as ATCO speech. This curated test set demonstrated substantial diversity, drawing from 383 unique video sources distributed across 34 distinct channels, thereby minimizing potential bias from channel-specific recording characteristics or speaker idiosyncrasies.

To assess the completeness of our content filtering approach, we also conducted an analysis to iden-

tify false negatives, which are samples relevant to ATC communication but incorrectly classified as irrelevant by the LLM content filter. We randomly sampled 200 transcripts from the rejected pool and manually evaluated their relevance to ATC operations. This analysis revealed that only 9 samples (4.5%) contained relevant ATC content, suggesting an approximate false negative rate of 5%. While this indicates that a small portion of relevant content is excluded from the dataset, we consider this error rate acceptable for our use case, as it represents a reasonable trade-off between dataset purity and completeness.

The manual transcription process was conducted by a domain expert with specialized air traffic control (ATC) knowledge according to established ATC transcription conventions. Specifically, the ICAO phonetic alphabet was normalized (e.g., "alpha" standardized to "alfa"), callsigns were segmented uniformly (e.g., "ryan air" transcribed as two distinct words), and specialized markers were applied systematically: unintelligible segments received *[unk]* tags, hesitation sounds were marked with *[hes]* tags, non-English greetings and farewells were explicitly identified, and standardized abbreviations (e.g., ILS, QNH) were preserved in their conventional forms.

Following transcription, we conducted a comprehensive quality assessment to characterize the nature and frequency of various audio quality issues inherent to YouTube-sourced ATC content. Crosstalk, where multiple speakers are present simultaneously, was identified in 30 files (1.8%), indicating cases where speaker diarization failed to correctly split the audio into individual speaker segments. Incomplete utterances appeared in 95 files (5.6%), reflecting errors in the voice activity detection model. These error rates are expected given the uncontrolled recording conditions typical of user-generated content. Background music contamination was notably prevalent, occurring in 148 files (8.7%), which is characteristic of virtual ATC streams where content creators provide musical entertainment for viewers. We regard this as non-critical since speech remains intelligible, and these samples may actually benefit model training by providing exposure to adverse acoustic conditions, functioning similarly to data augmentation techniques. Artificial voices, which are part of some games, were identified in 62 files (3.6%), representing synthetic speech from flight simulation software rather than authentic human communication. Non-English speech appeared in 51 files (3.0%), which reflects authentic ATC communication patterns where controllers occasionally interact with local pilots in non-English languages. Since our models are primarily tuned for English transcription, these segments may be inadvertently transcribed as ATC-like content and retained by the content filtering pipeline. Additionally, 281 files (16.5%) contained at least one [unk] tag, indicating that portions of these recordings were not entirely intelligible due to poor audio quality, heavy accents, or other acoustic challenges. YouTube-specific commentary presented a more nuanced challenge, occurring in 152 files (8.9%). While this content was not entirely unrelated to ATC operations, it contained clearances and phraseology that would not occur in authentic operational contexts. Nevertheless, these samples are not without value, as most contain domain-relevant terminology despite their non-operational nature. Following this quality screening process, 1,184 files remained suitable for detailed speaker role analysis. Manual classification of these files revealed 885 controller utterances and 299 pilot utterances, demonstrating that approximately 75% of utterances were correctly identified by the automated LLM-based speaker role classification system.

Finally, we quantified transcription accuracy by calculating the Word Error Rate (WER) between the automatic and manual transcripts, utilizing what is widely accepted as the de facto standard metric for ASR performance evaluation [27, 28]. We calculate the WER as follows:

$$WER = \frac{S + D + I}{N} \times 100\% \tag{1}$$

where:

- S = number of substitutions (words that were incorrectly recognized)
- D = number of deletions (words that were missed/not recognized)
- I = number of insertions (extra words that were incorrectly added)
- N = total number of words in the reference (ground truth) transcription

During this calculation, we exclude non-English and unintelligible parts of the transcript. Figure 8 presents the resulting WER for each approach, separated by speaker role.

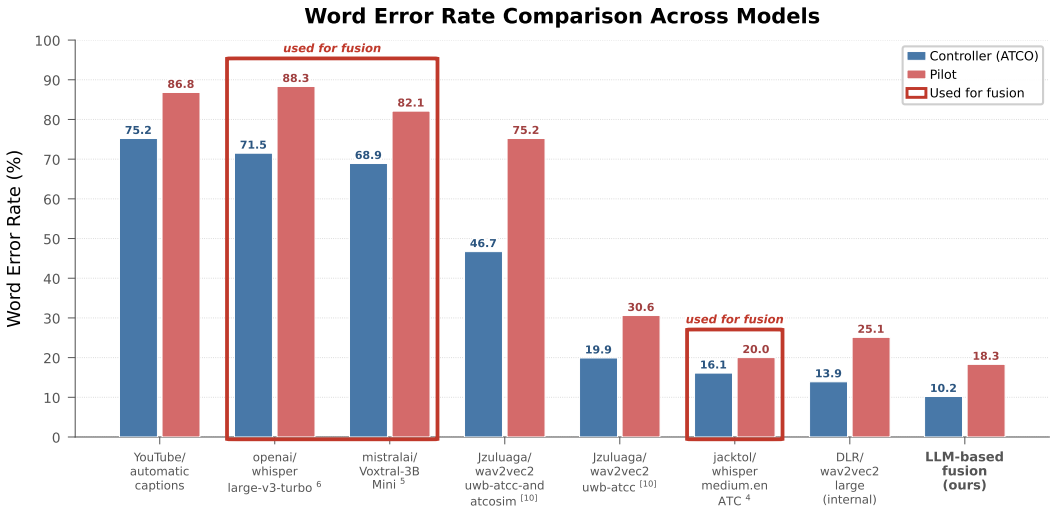

Figure 8. Calculated Word Error Rates for different models on our manually transcribed ground truth test set.

The ATC-specialized Whisper model[5] achieves 16.1% WER for controller speech and 20.0% for pilot speech. General-purpose models perform substantially worse: Voxtral-Mini[6] achieves 68.9% WER for controllers and 82.1% for pilots, while Whisper-large-v3-turbo[7] achieves 71.5% and 88.3%, respectively. These high error rates stem primarily from differences in transcription style. One example is number and letter transcription conventions: general-purpose models transcribe numbers using digits (i.e., FL280) while the ATC-tuned model transcribes the words verbatim (flight level two eight zero) which matches the style in our labeled ground truth. Our LLM-based fusion approach combines the three ASR outputs using Mistral Small 3.2[8], and achieves a substantial improvement: 10.2% WER for controllers (5.9 percentage points absolute improvement over the open-source ATC-specialized model) and 18.3% for pilots (1.7 percentage points improvement). This result demonstrates that complementary error characteristics across models can be effectively exploited through intelligent transcript combination, even when two of the three candidate models exhibit poor individual performance on this domain. The improvement is particularly remarkable considering the substantial domain mismatch of the general-purpose models. The LLM successfully identifies and selects the ATC-specialized model's output in most cases while occasionally correcting its errors using information from alternative transcriptions. This fusion approach enables the creation of large-scale training data with error rates suitable for downstream model training via knowledge distillation.

For comparison, we also calculated the word error rates based on the output of four other ASR mod-

[5]Model available at: https://huggingface.co/jacktol/whisper-medium.en-fine-tuned-for-ATC
[6]Model available at: https://huggingface.co/mistralai/Voxtral-Mini-3B-2507
[7]Model available at: https://huggingface.co/openai/whisper-large-v3-turbo
[8]Model available at: https://huggingface.co/mistralai/Mistral-Small-3.2-24B-Instruct-2506

els. The automatic capturing from YouTube itself achieved word error rates of 75.2% for controllers and 86.8% for pilots. Furthermore, there are three ASR models based on Facebook's Wav2Vec2.0 model architecture [29]. All of them have been fine-tuned with different ATC data. Wav2vec2-large-960h-lv60-self-en-atc-atcosim [10] has been fine-tuned with only the high-quality simulator dataset ATCOSIM. The model does not seem to be competitive on the tested data compared to the other two ATC-fine-tuned individual ASR models with word error rates of 46.7% for controllers and 75.2% for pilots. Wav2vec2-large-960h-lv60-self-en-atc-uwb-atcc-and-atcosim [10] has been fine-tuned with ATCOSIM and UWB-ATCC. This model achieves word error rates of 19.9% for controllers and 30.6% for pilots. The non-public model wav2vec2-large-internal has been fine-tuned with ATCOSIM, and approximately 100 hours of further internal English ATC communication data from operations and simulations including various accents. Although the word error rate of 13.9% for controllers is the best of all deployed models, the word error rate of 25.1% for pilots is only the second best. Still, the LLM-based fusion outperforms all individually deployed ASR models. To evaluate the performance on real-world ATC data, we train an NVIDIA NeMo model on our collected dataset. This process is described in detail in Section 5.

5. Feasibility for Transcription of Real ATC Data

The dataset of combined transcripts created earlier, consisting of 841 hours of audio with pseudo-labels classified as *ATCO*, *Pilot*, *unsure*, and *irrelevant*, were subsequently used to train a new ASR model. We aim to distill the knowledge contained in the combined transcripts into a small ASR model, as our multi-model transcription with LLM transcript fusion is too complex and too computationally expensive for real-time inference. The model was trained on all samples classified as *ATCO*, *Pilot*, or *unsure* to achieve a large-scale dataset, totaling 464 hours of training data. Only pseudo-labeled data was used during training; no other datasets were mixed into the training corpus. Instead of using a byte pair encoding tokenizer, as used in the Whisper models, we chose a character-based vocabulary to improve generalizability to new airlines or waypoints. We used the Nvidia Neural Modules (NeMo) Framework[9] to train a Fastconformer-Hybrid-RNNT Model using the fast Connectionist Temporal Classification (CTC) loss function. The trained model weights are publicly available on Hugging Face[10]. Compared to the other models presented, this model is the smallest, featuring 115M parameters (300M for Wav2Vec-large, 769M/1.55B for Whisper medium/large, and 3B parameters for Voxtral Mini Transcribe).

The model was trained on a single NVIDIA RTX A6000 Ada GPU for 67 epochs, totaling 70 hours of training time. We evaluate and compare the model to the model published by Zuluaga et al. [10], which was trained on a smaller corpus of high-quality, human-transcribed data consisting of UWB-ATCC and ATCOSIM. For evaluation, we use the ATCO2 test set, which contains 1 hour of data recorded from VHF radio. Our model outperforms the model proposed by Zuluaga et al., achieving a WER of 21.1% compared to 35.6% while being roughly 5× faster during inference. On the higher-quality ATCOSIM dataset, which was used during training of the baseline, our model achieves a WER of 14.9%. Considering the 10.2% WER measured on our human-transcribed test data, this result indicates that performance may be limited by the WER achieved during dataset creation. These results demonstrate that using our dataset consisting of pseudo-ATC and ATC-adjacent data from YouTube can significantly improve the transcription of real-world ATC communications. Furthermore, we expect that combining our YouTube-based training approach with established public ATC datasets such as ATCOSIM, UWB-ATCC, and LDC-ATCC would achieve WER values below 10%, which would enable many downstream applications in the ATC domain.

[9] https://github.com/NVIDIA-NeMo/NeMo
[10] https://huggingface.co/niclaswue/youtube-atc-fastconformer

Finally, we used our fine-tuned NeMo model to evaluate its ASR output on the test set shown in Figure 8. It should be taken into account that the fine-tuning data included this test set with the automatically generated LLM-based fusion transcripts. The NeMo-based model achieves a word error rate of 9.6% for controllers and 16.5% for pilots, which surpasses the 10.2% WER and 18.3% WER of the pseudo-labels themselves. This demonstrates that the model learned to average out errors in the training data and generalizes beyond the noisy labels by capturing underlying acoustic-linguistic patterns at scale.

6. Conclusion

This work directly addresses the question posed in our title: *Can YouTube stream recordings improve speech recognition for air traffic control?* Our results provide a clear answer: yes. By exploiting publicly available YouTube streams from virtual ATC simulator networks, we demonstrate substantial improvements in ASR performance despite the inherent domain shift between virtual and operational environments.

We developed an automated pipeline that processes over 2,000 hours of content, combining speaker diarization, multi-model transcription, and LLM-based transcript fusion to create a large-scale, diverse ATC-style dataset spanning multiple accents, regions, and operational domains. Additionally, we fine-tuned an ASR model exclusively on this automatically generated data and show that it can outperform models trained on fewer, higher-quality samples. We openly release the code for the processing pipeline, curated video collection, and our trained model to enable reproducible research.

Our key findings demonstrate:

1. **Dataset scale and diversity**: The collected dataset comprises 709 videos totaling over 2,000 hours from virtual airports in 17 countries, with substantial geographic coverage across multiple continents. The dataset spans multiple operational domains including ground, tower, approach, and en-route operations, providing comprehensive representation of real-world ATC scenarios.
2. **Automated transcription quality**: Our LLM-based fusion approach, combining outputs from three complementary ASR models, achieves 10.2% WER on controller speech and 18.3% WER on pilot speech in manually verified samples. This represents a substantial improvement of 5.9 percentage points over individual ATC-specialized open-source models, demonstrating that intelligent transcript fusion can effectively exploit complementary error characteristics across different models.
3. **Operational performance**: Despite training exclusively on pseudo-labeled YouTube data with no operational recordings, our distilled model achieves 21.1% WER on the operational ATCO2 benchmark, outperforming baseline models trained on smaller high-quality datasets (35.6% WER) while offering approximately five times faster inference speed. This substantial improvement demonstrates that YouTube-derived training data can effectively bridge the gap to real-world operational performance.

These results conclusively demonstrate that large-scale, diverse training data can effectively compensate for moderate label noise and domain shift between virtual ATC and real operational environments. The validity of this approach is further supported by our word frequency analysis (Figure 7), which reveals that word distributions in our ATCO and Pilot classes closely match those observed in real-world operational ATC communications, confirming substantial vocabulary and pattern overlap despite the source domain difference. Therefore, we believe this dataset could serve as valuable pretraining data for ATC ASR models, particularly when combined with smaller amounts of high-quality operational recordings for fine-tuning.

Future work should investigate iterative refinement strategies where trained models generate im-

proved pseudo-labels for subsequent training iterations, explore the integration of additional or improved ASR model outputs in the transcript fusion to further enhance transcription quality, extend the approach to multi-lingual ATC communications, and evaluate the dataset's utility for downstream tasks including semantic command parsing and context-aware speech understanding in operational ATC environments. Additionally, incorporating automated consistency checks for the LLM-based fusion, such as syntactic validators grounded in ATC grammar rules or multi-LLM consensus mechanisms, could further improve pseudo-label reliability. Preserving sequential context across adjacent utterance segments could enable the exploitation of discourse-level structure inherent in controller–pilot communication exchanges. Furthermore, extending the data collection to include real-world off-nominal ATC recordings from publicly available sources, potentially with tailored audio preprocessing, could provide complementary training data that further stress-tests model robustness under challenging acoustic conditions. Beyond aviation, we encourage researchers to adapt this methodology to other low-resource ASR domains where similar constraints of limited data availability prevent the development of robust speech recognition systems.

Author contributions

Conceptualization (N.W.), Data Curation (N.W., O.O.), Formal analysis (N.W., O.O.), Investigation (N.W.), Methodology (N.W.), Project administration (N.W., O.O.), Resources (N.W., O.O., M.K.), Software (N.W., M.K.), Supervision (N.W, O.O.), Validation (O.O.), Visualization (N.W.), Writing – Original Draft (N.W., O.O.), Writing – Review & Editing (N.W., O.O., M.K.).

Open data statement

The dataset generated and analyzed in this study is publicly available at https://github.com/niclaswue/youtube-atc.

Reproducibility statement

To ensure reproducibility, we provide the complete processing pipeline code, curated video collection, trained model weights, and detailed instructions at https://github.com/niclaswue/youtube-atc. Additionally, the model weights for the model trained in this paper can be directly accessed and downloaded via Hugging Face at https://huggingface.co/niclaswue/youtube-atc-fastconformer.

References

[1] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. "Robust speech recognition via large-scale weak supervision". In: ICML'23. Honolulu, Hawaii, USA: JMLR.org, 2023.

[2] Hanada Said. "Pilots/Air Traffic Controllers Phraseology Study". In: *International Air Transport Association* (2011).

[3] Luboš Šmídl, Jan Švec, Daniel Tihelka, Jindřich Matoušek, Jan Romportl, and Pavel Ircing. "Air traffic control communication (ATCC) speech corpora and their use for ASR and TTS development". In: *Language Resources and Evaluation* 53.3 (2019), pp. 449–464. DOI: 10.1007/s10579-019-09449-5.

[4] John J. Godfrey. *Air Traffic Control Corpus (ATC0)*. LDC Catalog Number LDC94S14A. ISBN 1-58563-024-1. 1994. DOI: 10.35111/mhz2-w697.

[5] Konrad Hofbauer, Stefan Petrik, and Horst Hering. "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)*. Marrakech, Morocco: European Language Resources Association (ELRA), 2008. URL: https://aclanthology.org/L08-1507/.

[6] Juan Zuluaga-Gomez et al. *ATCO2 corpus: A Large-Scale Dataset for Research on Automatic Speech Recognition and Natural Language Understanding of Air Traffic Control Communications*. arXiv:2211.04054. 2022.

[7] Xinjian Li, Shinnosuke Takamichi, Takaaki Saeki, William Chen, Sayaka Shiota, and Shinji Watanabe. "Yodas: Youtube-Oriented Dataset for Audio and Speech". English. In: *2023 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2023*. 2023 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2023. Publisher Copyright: © 2023 IEEE.; 2023 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2023 ; Conference date: 16-12-2023 Through 20-12-2023. Institute of Electrical and Electronics Engineers Inc., 2023. DOI: 10.1109/ASRU57964.2023.10389689.

[8] Nithin Rao Koluguri, Monica Sekoyan, George Zelenfroynd, Sasha Meister, Shuoyang Ding, Sofia Kostandian, He Huang, Nikolay Karpov, Jagadeesh Balam, Vitaly Lavrukhin, et al. "Granary: Speech Recognition and Translation Dataset in 25 European Languages". In: *arXiv preprint arXiv:2505.13404* (2025).

[9] Alexander H. Liu et al. *Voxtral*. 2025. arXiv: 2507.13264 [cs.SD]. URL: https://arxiv.org/abs/2507.13264.

[10] Juan Zuluaga-Gomez, Amrutha Prasad, Iuliia Nigmatulina, Seyyed Saeed Sarfjoo, Petr Motlicek, Matthias Kleinert, Hartmut Helmke, Oliver Ohneiser, and Qingran Zhan. "How Does Pre-trained Wav2Vec 2.0 Perform on Domain Shifted ASR? An Extensive Benchmark on Air Traffic Control Communications". In: *2022 IEEE Spoken Language Technology Workshop (SLT)*. Doha, Qatar: IEEE, 2023, pp. 205–212. DOI: 10.1109/SLT54892.2023.10022724.

[11] Holger Schwenk and Jean-Luc Gauvain. "Combining multiple speech recognizers using voting and language model information." In: *INTERSPEECH*. 2000, pp. 915–918.

[12] Jiahao Fan and Weijun Pan. "Customization of the ASR System for ATC Speech with Improved Fusion". In: *Aerospace* 11.3 (2024). ISSN: 2226-4310. DOI: 10.3390/aerospace11030219. URL: https://www.mdpi.com/2226-4310/11/3/219.

[13] Chen Chen, Ruizhe Li, Yuchen Hu, Sabato Marco Siniscalchi, Pin-Yu Chen, Ensiong Chng, and Chao-Han Huck Yang. "It's Never Too Late: Fusing Acoustic Information into Large Language Models for Automatic Speech Recognition". In: *International Conference on Learning Representations (ICLR)*. arXiv:2402.05457. 2024.

[14] Chan-Jan Hsu, Yi-Chang Chen, Feng-Ting Liao, Pei-Chen Ho, Yu-Hsiang Wang, Po-Chun Hsu, and Da-shan Shiu. "Let's Fuse Step by Step: A Generative Fusion Decoding Algorithm with LLMs for Robust and Instruction-Aware ASR and OCR". In: *Findings of the Association for Computational Linguistics: ACL 2025*. arXiv:2405.14259. 2025, pp. 24959–24973. DOI: 10.18653/v1/2025.findings-acl.1281.

[15] Jeena Prakash, Blessingh Kumar, Kadri Hacioglu, Bidisha Sharma, Sindhuja Gopalan, Malolan Chetlur, Shankar Venkatesan, and Andreas Stolcke. *Better Pseudo-labeling with Multi-ASR Fusion and Error Correction by SpeechLLM*. arXiv:2506.11089. 2025.

[16] Chen Chen, Yuchen Hu, Chao-Han Huck Yang, Sabato Marco Siniscalchi, Pin-Yu Chen, and Eng Siong Chng. "HyPoradise: An Open Baseline for Generative Speech Recognition with Large Language Models". In: *Advances in Neural Information Processing Systems 36 (NeurIPS 2023), Datasets and Benchmarks Track*. arXiv:2309.15701. 2023.

[17] Sheng Li, Chen Chen, Chin Yuen Kwok, Chenhui Chu, Eng Siong Chng, and Hisashi Kawai. "Investigating asr error correction with large language model and multilingual 1-best hypotheses". In: *Proc. Interspeech*. Vol. 2024. 2024, pp. 1315–1319.

[18] Rithik Sachdev, Zhong-Qiu Wang, and Chao-Han Huck Yang. "Evolutionary prompt design for llm-based post-asr error correction". In: *arXiv preprint arXiv:2407.16370* (2024).

[19] Anu Pradhan, Alexandra Ortan, Apurv Verma, and Madhavan Seshadri. *LLM-as-a-Judge: Rapid Evaluation of Legal Document Recommendation for Retrieval-Augmented Generation*. 2025. arXiv: 2509.12382 [cs.CL]. URL: https://arxiv.org/abs/2509.12382.

[20] Kye Taylor. *2024 - Statistics and Review*. VATSIM UK. Jan. 2025. URL: https://community.vatsim.uk/blogs/entry/584-2024-statistics-and-review/ (visited on 10/27/2025).

[21] Jack Tol. *Fine-Tuning Whisper for Air Traffic Control: 84% Improvement in Transcription Accuracy*. Oct. 2024. URL: https://jacktol.net/posts/fine-tuning_whisper_for_atc/ (visited on 10/27/2025).

[22] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. "Efficient Memory Management for Large Language Model Serving with PagedAttention". In: *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*. 2023.

[23] Mistral AI. *Mistral Small 3.2*. https://docs.mistral.ai/models/mistral-small-3-2-25-06. 24B parameter multimodal language model with vision capabilities. 2025.

[24] Sander Schulhoff et al. *The Prompt Report: A Systematic Survey of Prompt Engineering Techniques*. 2024. arXiv: 2406.06608 [cs.CL]. URL: https://arxiv.org/abs/2406.06608.

[25] Shuo Chen, Hartmut Helmke, Robert M Tarakan, Oliver Ohneiser, Hunter Kopald, and Matthias Kleinert. "Effects of language ontology on transatlantic automatic speech understanding research collaboration in the air traffic management domain". In: *Aerospace* 10.6 (2023), p. 526.

[26] Mark Davies. "The 385+ million word Corpus of Contemporary American English (1990–2008+): Design, architecture, and linguistic insights". In: *International journal of corpus linguistics* 14.2 (2009), pp. 159–190.

[27] Rahhal Errattahi, Asmaa El Hannani, and Hassan Ouahmane. "Automatic speech recognition errors detection and correction: A review". In: *Procedia Computer Science* 128 (2018). 1st International Conference on Natural Language and Speech Processing, pp. 32–37. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2018.03.005.

[28] Nicholas Ruiz and Marcello Federico. "Assessing the impact of speech recognition errors on machine translation quality". In: *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas: MT Researchers Track*. Association for Machine Translation in the Americas, 2014, pp. 261–274.

[29] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. "wav2vec 2.0: A framework for self-supervised learning of speech representations". In: *Advances in neural information processing systems* 33 (2020), pp. 12449–12460.