Twelfth USA/Europe Air Traffic Management Research and Development Seminar (ATM2017)

Read Back Error Detection using Automatic Speech Recognition

Shuo Chen, Hunter Kopald, Dr. Ronald S. Chong, Dr. Yuan-Jun Wei, Zach Levonian
Center for Advanced Aviation System Development (CAASD)
The MITRE Corporation
McLean, Virginia
chen@mitre.org

***Abstract*—In the last few years, the Federal Aviation Administration (FAA) has been investigating the use of automatic speech recognition in safety monitoring capabilities, with an initial focus on the tower domain. One application of speech recognition technology is the automatic detection of pilot read back errors that are not corrected by the controller. Uncorrected read back errors are one cause of runway incursions, one of the FAA's primary surface safety concerns. To inform the FAA's investigation into future speech recognition applications and mitigate the risk associated with future capability development, the MITRE Corporation (MITRE) is conducting research into the feasibility of a read back error detection capability that uses speech recognition to compare controller and pilot intent spoken in radio transmissions from the tower domain. Through this research, MITRE has developed a concept of use for the capability that includes a high level system design and a graphical user interface design. Additionally, MITRE conducted an evaluation of speech recognition performance on controller and pilot radio transmissions to assess the accuracy achievable with established speech recognition technology and tuning methods. Preliminary results from the research indicate that speech recognition performance on controller and pilot speech is promising, but more research is needed to refine the capability logic, improve speech recognition accuracy, and assess operational acceptability of its performance.**

Keywords: read back error detection, automatic speech recognition, safety

I. INTRODUCTION

Radio transmissions remain the primary means of communication between controllers and pilots in the Air Traffic Control (ATC) domain. While a Data Communication, or text-based transmission of data, between controllers and pilots is envisioned to supplement radio communications in future operating environments, this capability is unlikely to completely replace radio communications in the near term. Controllers ensure the safety of flights operating within their jurisdiction by issuing clearance commands and advisories to pilots, verifying the pilots' immediate read back of the commands for correctness and understanding, and then continually monitoring aircraft movement to ensure compliance to issued commands.

In the airport surface environment, local and ground controllers are responsible for ensuring the safe maneuvering of arriving, departing, and taxiing aircraft in and around a towered airport. During busy periods, controllers may be responsible for many aircraft simultaneously, with a significant amount of their time and workload devoted to voice communications and read back monitoring.

In the last few years, the Federal Aviation Administration (FAA) has been investigating the use of automatic speech recognition to provide or support automated monitoring capabilities that can improve surface safety. The research has focused on demonstrating feasibility with a simple use case that recognizes a constrained set of phrases in only controller-side transmissions because the regulated phraseology, limited speaker set, and cleaner acoustic quality make controller transmissions more conducive to speech recognition than their complement—pilot transmissions. However, in parallel with a demonstration of this initial controller speech-based surface monitoring prototype at John F. Kennedy International Airport (JFK), the FAA is researching more complex speech-based applications that offer more potential benefit by detecting a larger number of safety situations [1][2].

With an eye toward future capabilities for the use of speech recognition in real-time air traffic operations The MITRE Corporation's Center for Advanced Aviation System Development (MITRE CAASD) is investigating the feasibility of a read back error detection capability that uses speech recognition to compare the content of controller and pilot transmissions. To assess the feasibility of speech recognition technology for a given application, the application logic must be defined and the speech recognition performance must be measured against the application's needs. For read back error detection, the application's needs are defined by the logic (rules) for what warrants an alert. The speech recognition performance must then be measured in the context of this logic—in other words, how well the speech recognition system correctly recognizes the speech information needed to apply the logic to generate the desired alerts.

Section II provides background information on read back errors in the ATC domain and on automatic speech recognition technology, including its application to ATC speech. Section III describes the concept, an initial user interface design, and the first steps toward deriving alert logic. Section IV describes the speech recognition performance analysis, including performance results on both controller and pilot speech. Section V presents the next steps toward developing a proof-of-concept demonstration system.

II. Background

A. *Read back Errors in ATC Communications*

A substantial portion of a controller's job is issuing commands to aircraft. After each clearance or instruction, the controller is to "ensure that items read back are correct, ensure the read back of hold short instructions…, and ensure pilots use call signs and/or registration numbers in any read back acknowledging an air traffic clearance or ATC instruction" [3]. At times, pilots may make a read back error. The misunderstanding may then be compounded if the controller does not correct the error in the pilot read back—a so called hear back error. The resulting uncorrected read back error can put the operation in an unsafe state since it is unknown if the pilot will comply with the clearance or instruction (in which case, the pilot simply misspoke during the read back) or, if the pilot misheard the clearance or instruction and their read back indicates the action they will actually perform. While a read back error and subsequent hear back error concerning an instruction to taxi in to the ramp may not have serious consequences, a read back and subsequent hear back error concerning runway use could have significant safety implications.

The research team reviewed the literature on controller-pilot communication errors to better understand the prevalence of the problem of read back and hear back errors. In a series of papers reporting on controller-pilot voice communications data from the tower domain (both local and ground positions), in 1993 and 1996, Cardosi found pilot read back error rates of ≤ 1% and that 40% of read back errors were uncorrected, i.e., hear back errors [4][5]. Cardosi reported similar results for the En Route and Terminal Radar Approach Control (TRACON) domains [6][7]. Cardosi et al. reported that in these earlier studies, the number of read back errors was about one per hour per frequency across all domains [8]. A more recent study which examined only the TRACON reported read back error rates of 6% with a hear back error rate of 92% [9]. The authors note that differences in analysis methodology relative to previous studies may explain the significant difference in read back and hear back error rates. Eurocontrol conducted similar research, studying incidents in Europe and adopting Cardosi's taxonomy, and found that read back/hear back errors were the most common controller-pilot communication problem [10].

None of these studies provided an estimate of the probability of an incident due to a read back/hear back error. As the focus of this research is on the tower domain, runway incursions, situations in which an unauthorized aircraft is on a runway, are a key potential consequence of communication errors. Kopald and Goring analyzed runway incursions attributed to controller error (i.e., operational incidents) and identified runway incursions where a hear back error was likely a contributing factor [11]. The analysis found that read back errors accounted for 10.7% of the runway incursions in a subset of a 6-year runway incursion dataset. The analysis extrapolated that percentage to 129 runway incursions in the full dataset due to read back errors.

Using the aforementioned estimate of one read back error per hour per frequency, along with other factors—40% of which are hear back errors, assuming an average of two frequencies per tower, for ~500 towers in the National Airspace System (NAS), and 129 runway incursions attributed to hear back errors over the 6-year period—it is clear that the absolute risk of a runway incursion due to a hear back error is vanishingly small. Specifically, there is a risk of 1 runway incursion for every 163,000 hear back errors, 1 runway incursion for every 407,000 read back errors, or 1 runway incursion for every 40,700,000 commands. Although this probability of an incursion is very small, the potential consequence of even a single runway incursion might necessitate capabilities to mitigate that risk.

Despite the rare occurrence of read back/hear back errors, current and retired air traffic controllers and managers frequently suggest that a read back error detection capability would be a useful application of automatic speech recognition technology. Given that evaluating pilot read back is a significant portion of a controller's responsibilities and given the potential negative consequences of a hear back error, it is understandable that controllers and managers would be enthusiastic about the potential benefit of a capability that could (1) detect and alert for pilot read back error and (2) detect and alert when the controller has not provided a timely corrective command in response to a pilot read back error, thereby preventing hear back errors. The key challenge to ensuring such a capability is useful is to define logic that alerts only for the appropriate situations.

B. *Automatic Speech Recognition Technology*

Automatic speech recognition (ASR) translates digitized audio to a text transcription of the speech content. Speech recognition systems typically employ both an acoustic model (AM) and a language model (LM) in tandem to produce one or more transcription hypotheses with an associated likelihood of the transcription's accuracy [12]. The AM breaks down a digitized audio sample into a sequence of likely phonemes, which are basic units of speech sounds [13]. The LM accepts the sequence of phonemes and combines them to form a probable sequence of words.

Statistical methods are used to create LMs and AMs, prior to recognition, from labeled speech data [13][14]. Both types of models rely on large quantities of training data to create and tune [15]. The accuracy and robustness of these statistical models are dependent on the amount of training data available and on how well the training data matches the data that will be recognized in the operational environment. Thus the collation of appropriate training data is important to the performance of a speech recognition system for a specific application.

Speech in the ATC domain varies from typical conversational English in several important ways. Speech in the ATC domain is highly specialized, with many domain-specific terms, and is typically faster than conversational speech. ATC phraseology for controllers is designed to be concise with little repetition or redundancy. Phraseology for pilots is less regulated but tends to be even more terse than controller speech, making it difficult to understand without situational or dialogue context. The voice switches used to transmit and receive radio transmissions between controllers and pilots introduce acoustic characteristics that are unique to the domain.

Previous work involving speech recognition in the ATC domain has focused primarily on training the LM, by adjusting or constraining it, to improve ASR performance, with less focus on tuning the AM [16][17][18][19]. However, the limited vocabulary of ATC speech, the variation in speakers and speech environments, and the physical channel characteristics all suggest that greater changes in the AM can improve recognition accuracy.

AMs can be tuned for the unique conditions of a particular application either through training of a task-specific acoustic model or adaptation of a pre-existing generic acoustic model [13]. Training data is needed for both training an acoustic model and adapting an acoustic model. Training an acoustic model requires a large amount of training data to ensure good accuracy and model stability. Adaptation, on the other hand, uses a smaller amount of task-specific data to adapt (i.e., modify) a pre-existing, larger, and more general acoustic model to the new environment. Even with limited training data, adaptation can still account for variations in the recording equipment, for new speakers, and for other features that vary from the acoustic model training data to the operational environment [20]. While there exist different approaches to adaptation, each approach adjusts the statistical probabilities within the models without changing the fundamental structure of the model itself [21][22][23][24].

Although automatic speech recognition has become increasingly prevalent in the ATC domain, most of the work has focused on recognizing controller-side communications for the purposes of improving safety during live operations, facilitating flight data entry, standardizing controller training, providing automated simulation pilot capabilities in lab environments, and augmenting post-event analysis. A read back error detection capability differs from these previous applications in that it requires correct recognition of both controller and pilot transmissions to function properly. The concept of an automatic read back error detection capability is not new to the domain—Ragnirsdottir et al. proposed a language technology system that could support read back error detection in the oceanic environment as early as 2003 [25]. However, research into the feasibility of a real-time, read back error detection capability has been limited, partly because of the difficulty associated with recognizing pilot speech. This research aims to assess the feasibility of recognizing both controller and pilot speech for an automated read back error detection capability, with a focus on quantifying the benefits of AM tuning through adaptation and training.

III. READ BACK ERROR DETECTION CONCEPT AND DESIGN

A. Capability Concept and Design

The read back error detection capability envisioned is simple and intuitive. A speech recognition system passively monitors controller and pilot radio transmissions. Aircraft identifiers are identified within transmissions and used to match controller and pilot transmissions as command-read back pairs. Clearance commands and read backs are parsed to extract their meaning for comparison. When a mismatch between a command and a read back transmission pair is detected or when a read back is altogether absent, the system generates an alert for the controller. The system can be extended by expanding the transmission pairing beyond two transmissions to allow for corrective transmissions that the controller issues without any prompting because the read back mismatch was already detected by the controller. Furthermore, the context and content of the transmissions can be used as supporting or alternative criteria for matching transmissions, in case an aircraft identifier is not spoken as part of a transmission or was not recognized by the speech recognition component. Figure 1 depicts the high level components of the capability.

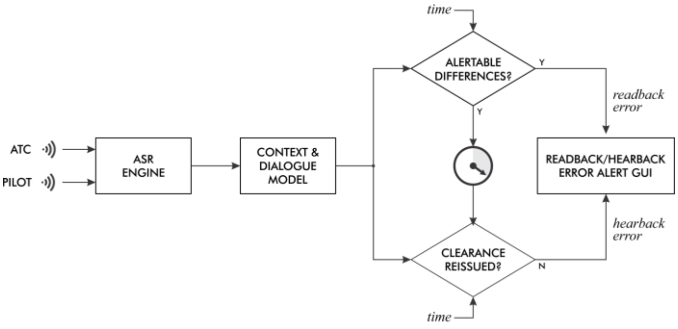


Figure 1. High level components of the read back error detection capability.

B. Graphical User Interface

A graphical user interface (GUI) for the read back error detection system—an interface where alerts would be displayed and managed—was designed with input from subject matter experts (SMEs). The information elements that could be included in the GUI are described below and then the layout of the GUI itself is presented.

A number of basic information elements could be presented through a GUI, including:

- Raw ASR result of controller and/or pilot utterance: The raw text output of the speech recognition engine
- Concise speech recognition result of controller and/or pilot utterance: The processed output of the ASR engine that provides the aircraft identifier, semantic and informational elements of the utterance
- Speech recognition confidence per utterance: A dimensionless scalar number indicating the speech recognition engine’s confidence in its transcription.
- Dialogue timeline/history: a chronological history of all pilot and controller transmissions

- Speech recognition system status/activity: indicators that show that the system is operating, recognizing who was just speaking, and is processing the utterances.

- Cause of the alert: an indicator that shows an explanation of why the alert occurs; e.g., a missing read back, a broken transmission, or errors in the read back.

- An advisory command that could be issued by the controller to respond to the alert

- Playback of the recorded controller and pilot audio triggering an alert

- An audible alert embodied as a brief, non-repeating tone or a spoken alert

Additionally, with respect to user interaction, alerts could be actively acknowledged by a controller or passively cleared by the system when certain conditions are met, such as:

- The controller issues the same command to the aircraft

- The controller issues a different command to the aircraft in response to or in spite of a read back error

- Occurrence of a new read back error detection alert for any aircraft; in order to prevent simultaneous alerts

- A pre-configured amount of time has transpired since the alert; useful for a slow environment

- A pre-configured number of controller utterances have occurred since the alert; useful for a busy environment

These preliminary lists of information elements and interaction criteria were presented to ten SMEs, MITRE staff who are retired controllers, as a starting point in GUI design discussions. The goal of the discussions was to deduce a list of essential information elements and interactions that the GUI should support. Some elements, such as the advisory command, were quickly dismissed during the discussion because the SMEs felt that this advisory removed too much of the decision making discretion from the controller. Other elements, such as having the system wait to issue an alert in case the controller takes corrective action, were well received. In the end, the general consensus was that read back error detection GUI should provide the following basic information elements:

- Aircraft identifier

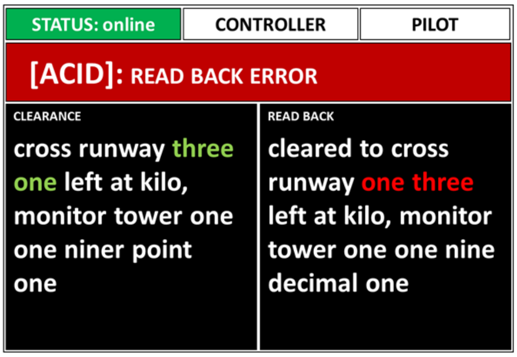
- An indication that a read back is missing, where appropriate: A few SMEs believed this visual element was not necessary

- An indication that a read back error occurred, where appropriate: A few SMEs believed this visual element was not necessary

- Indication/s of the difference/s between the command and read back: displaying both the command and the read back, in the concise form, with an indication of the differences.

- Audible alert: The SMEs agreed that an audible alert indicator would be needed but disagreed whether the alert should be only in the ear of the affected controller or a public alert in the tower.

Figure 2 depicts a notional design for the read back error detection GUI that contains the information elements identified by SMEs as necessary and sufficient. It illustrates the visual depiction of a read back error. The difference between the command and the read back in this example is shown by a color difference. If the controller does not correct the command within a certain time delay, then a brief, non-repeating auditory alert would be presented to draw the controller's attention to the display. The read back error detection logic would then automatically reset to the non-alert condition according to reset criteria described earlier.



STATUS: online | CONTROLLER | PILOT

[ACID]: READ BACK ERROR

CLEARANCE

cross runway three one left at kilo, monitor tower one one niner point one

READ BACK

cleared to cross runway one three left at kilo, monitor tower one one nine decimal one

Figure 2. Notional design of the read back error detection GUI.

C. *Monitoring and Alert Design*

Read back errors may initially be thought of as the presence of a difference in semantics and information content between the controller's command and the pilot's read back. Certainly, if a read back perfectly echoes the command, it is easy to know that no read back error has occurred. Inversely, if a read back is not provided, it is easy to know that a read back error has occurred. However, the vast majority of read backs fall between these two extremes: differences (along various dimensions) exist, but the controller need not issue a corrective command to confirm pilot understanding. So a challenge of this research is to determine, from the controller's perspective, a) which differences are alert worthy and which differences are not, and b) does this characterization change as a function of the operational context (e.g., current workload, which runways are open, position/s being worked).

As 99% of controller commands do not result in read back errors, pilot-controller dialogues can be analyzed to identify what differences exist that controllers do and do not consider read back errors. Using this approach, it may be possible to empirically derive an effective and feasible alerting principle that can distinguish which pilot read back differences should and should not be ignored.

An analysis of 150 dialogues (drawn from a much larger dataset) between ground control and pilots at John F. Kennedy International Airport (JFK) in New York, USA was conducted. Here's an example of a pilot-controller exchange that did not prompt corrective command from the controller (aircraft call signs have been anonymized):

Controller: [CALLSIGN], Kennedy ground, three one left kilo echo, taxi right alfa, hold short of juliet.

Pilot: And alfa short of juliett, for three one left kilo echo, [CALLSIGN].

A difference commonly seen between commands and read backs, particularly when commands are lengthy, is the ordering of the elements, as seen here. The controller specified a right turn at alfa, but the pilot read back did not include the turn or direction and this difference was uncorrected. A possible reason may be that the controller is aware that the airport diagram is present on the flight deck and that the crew know their destination (it was correctly acknowledged in the read back) and it is clear that the only/fastest way to reach the destination was to make a right at alfa.

Here is another example where the controller did not correct the differences in the read back:

Controller: [CALLSIGN-0] at kilo, follow [CALLSIGN-1] off the right, cross runway three one left, and monitor tower one niner point one.

Pilot: Okay, behind the [CALLSIGN-1] five seven from the right, follow her down kilo for zero four left, and nineteen one to monitor, [CALLSIGN-0]

The semantics of the cross and follow are present in the read back but require some interpretation; i.e., the pilot did not repeat the controller's crossing instruction (which mentioned runway 31L) but did repeat the controller's instruction to follow another aircraft to a runway (mentioning 4L). The transfer of control to the tower and on a particular frequency is also present in the read back but in a very abbreviated and implied form. The command included an explicit runway crossing instruction but there was only an implicit acknowledgement through the phrase "follow her down kilo for zero four left". None of these differences elicited a correction from the controller. This example illustrates that one of the technical challenges of a read back/hear back error detection capability is both extracting explicit and deducing/inferring implicit semantic information from read backs and combining it with other data sources (e.g. runway and taxiway layout in this case).

Here's a final example of a dialogue that does include corrective commands:

Controller: [CALLSIGN], cross runway three one left at kilo, and monitor tower one niner point one.

Pilot: Roger, cleared to cross runway one three left at kilo, and then monitor tower one one nine decimal one, [CALLSIGN].

Controller: [CALLSIGN], cross runway three one left at kilo, and monitor tower one niner point one.

Pilot: [Uh] Roger, cross runway one three left at kilo, and then monitor tower one one nine one, [CALLSIGN].

Controller: [CALLSIGN], its runway three one left at kilo, cross runway three one left at kilo.

Pilot: Sorry, runway three one left at kilo then, my apologies, [CALLSIGN], thank you.

In the set of 150 dialogues, two additional dialogues contained read back differences that prompted the controller to require a correct read back. In one case, the taxiway name was incorrect; "kilo golf" vs "kilo". In the other, the handoff frequency was incorrect; 123.9 instead of 119.1.

The analysis identified potentially useful patterns concerning a) read back differences that controllers tolerate, b) behavior that is customary for controllers, and c) behavior that is customary for pilots. These behavioral patterns are:

- Order of elements: Controllers do not require the order of elements in read backs to match the command.
- Transfer of control/frequency change: Controllers do not require the read back to contain the facility name and/or frequency. (Simply saying "See ya" would suffice as acknowledgement of the transfer of control.) If an incorrect facility name and/or an incorrect full or abbreviated frequency is present in the read back, then the controller will correct the error.
- Callsigns: Pilots almost always used their full callsign (e.g., "American one two three"). For the few examples where pilots used only the carrier (e.g., "American") or the flight number ("one two three"), or provided no ID at all, controllers did not require a full or partial callsign.
- Taxi commands: Controllers consistently provided left/right turn specification in taxi instructions. Pilots regularly dropped the left/right turn specification in taxi commands. Controllers did not require a correct read back in these cases.
- Follow: Pilots almost always acknowledged the "follow" instruction with something semantically equivalent (e.g., "after", "behind").
- Crossing instructions: Controllers almost always used the form "cross <runway> at <intersection>" in their commands. Pilots always provided some form of read back acknowledging the crossing instruction. Controller do not require a complete read back of the crossing instruction; e.g., "cross, tower on the other side" would suffice.
- Hold Short: Pilots almost always correctly read back both the hold short instruction and position. Controllers may always require a correct and complete read back to this command given the potential safety risk an aircraft's untimely entry to an active runway and the fact that ensuring hold short instructions are read back is called out specifically in JO7110.65 [3].

The research team can start to develop an alerting principle that states: alert for differences that are a) beyond what controllers will tolerate, b) beyond what pilots customarily produce, and c) objectively unsafe. This alerting principle will be amended as more ground dialogues and, later, local (runway) controller dialogues are analyzed.

IV. Speech Recognition Evaluation and Results

The MITRE team created and compared several different automatic speech recognition configurations on the open-source engine, PocketSphinx, to evaluate the feasibility of using speech recognition in a read back error detection capability. Although newer and more advanced, open-source speech recognition engines that support cutting-edge techniques such as Deep Neural Network (DNN) based speech recognition exist, the research team elected to use PocketSphinx for this preliminary performance evaluation because of its ready-to-use, simple yet flexible application interface, processing speed, and compatibility with custom acoustic and language models [26].

The different types of models created for the comparisons are described below, along with the evaluation data and methodology. The section closes with a representative subset of the performance comparison results and findings.

A. Acoustic Models

Three types of acoustic models were evaluated:

1) A base acoustic model released by CMUSphinx for US English. The CMUSphinx base US English acoustic model is created using high quality microphone, broadcast, and telephone speech recordings and optimized for general large vocabulary applications [27]. It was deliberately left un-tuned for ATC speech to serve as a baseline of comparison to our adapted and trained models.

2) Adapted acoustic models based on CMUSphinx's base US English acoustic model and adapted using transcribed audio data from ATC operational recordings that contained both controller and pilot radio transmissions. Acoustic model adaptation, which may consist of maximum a posteriori (MAP) adaptation, maximum likelihood linear regression (MLLR), or some combination of both, is known to effectively adjust generic acoustic models for specific recording environments, audio channels, and slight accent differences. When used on an existing stable acoustic model and with limited training data, adaptation is more robust than training an acoustic model from scratch [27]. Both MAP and MLLR adaptation were used for this evaluation.

3) Trained acoustic model created using only silence-reduced, transcribed audio data from ATC operational recordings that contained both controller and pilot transmissions. Acoustic model training requires more time and data to perform, but stable, trained acoustic models can perform better than adapted acoustic models when the training data are sufficient in quantity and closely match the audio that will be recognized [28]. Additionally, acoustic models trained using only ATC data are better optimized in terms of both accuracy and speed for the constrained vocabulary in the ATC domain than the base CMUSphinx acoustic model, which was designed for large vocabulary applications.

Recordings from a variety of US tower and TRACON facilities, including Boston Logan International Airport (KBOS), John F. Kennedy International Airport (KJFK), Washington Dulles International Airport (KIAD), Reagan National Airport (KDCA), Dallas/Fort Worth International Airport (KDFW), Hartsfield-Jackson Atlanta International Airport (KATL), and Atlanta Terminal Radar Approach Control (A80), were used in the training data set. Furthermore, both pilot and controller radio transmissions were included to provide a sufficient quantity of data for model training stability.

B. Language Models

PocketSphinx is compatible with different types of language models, including keyword lists, grammars, and statistical language models (SLM). However, for this use case, the MITRE team favored SLMs over the other types of models because of the variability in the target speech, particularly on the pilot side. Two SLMs were created for the evaluation:

1) An SLM created using transcriptions of local and ground controller radio transmissions from KJFK for recognition on controller radio transmissions.

2) An SLM created using transcriptions of pilot radio transmissions heard at the local and ground controller positions at KJFK for recognition on pilot radio transmissions.

The research team decided that separating the pilot and controller transmissions was logical for training the language model during this initial performance evaluation because the differences in phraseology based on role (i.e., controller or pilot) are more significant than other criteria such as control position (i.e., ground or local), time of day, or phase of flight of the aircraft.

C. Semantic Meaning Extraction Algorithm

Because pilot read backs do not always exactly echo the controller command, a read back error detection capability must compare the semantic intent in the controller's commands with the semantic intent the pilot's read backs, rather than the exact words spoken by each party. The research team employed a semantic meaning extraction algorithm to parse this semantic intent, in the form of meaningful command and read back concepts, from the transcription hypotheses returned by the recognition engine. The algorithm identified command concepts in terms of a command phrase followed by one or more parameters phrases. For example, for the spoken command, "cleared to land runway two three", the algorithm would identify "cleared to land" as the command phrase and "runway two three" as the single parameter associated with that phrase. In the case of a compound taxi command such as, "continue on foxtrot alfa, turn right at bravo, taxi bravo to the ramp", the algorithm would parse "continue", "turn right", and "taxi" as the command phrases of three distinct command concepts and "foxtrot alpha", "alfa", and "bravo, ramp" as the parameters associated with each concept, respectively. In the case of the final "taxi" command concept, the algorithm would treat "bravo" and "ramp" as two separate, sequential parameters, both associated with the "taxi" command phrase.

For this evaluation, the semantic meaning extraction algorithm was configured to identify the limited set of command concepts listed in Table I. The algorithm accommodates small, logical variations in the command and parameter phrases such as tense differences ("clear" instead of "cleared"), omissions ("line up wait" instead of "line up and wait"), and literal alternatives ("one niner left" instead of "one nine left").

TABLE I. COMMAND CONCEPTS

Command Concept	Nominal Command Phrase	List of Parameter Alternatives
CTL	"cleared to land"	Runway
LUAW	"line up and wait"	Runway
CFT	"cleared for takeoff"	Runway
Hold Short	"hold short"	Runway or Taxiway
Cross	"cross"	Runway or Taxiway
Turn	"turn left" or "turn right"	Runway or Taxiway
Taxi	"taxi"	Runway(s) or Taxiway(s),
Continue	"continue"	Runway(s) or Taxiway(s)

Aircraft identifiers (ACID) are another semantic component that must be identified in order to determine the target of a controller command and the identity of the speaker in a pilot read back. This semantic information enables the read back error detection capability to correctly associate a controller command with a subsequent pilot read back. To identify ACID concepts, the research team employed a separate pattern matching algorithm that extracted ACID phrases in the recognized text, such as "southwest twenty-three seventy-four", and mapped them to their symbolic form, such as "SWA2374". The different spoken formulations associated with aircraft in the test data were automatically generated and input into the pattern matching algorithm as a reference list. The algorithm was then applied to the text returned by the recognition engine to extract any aircraft identifier phrases.

D. Test Data

The test data used for benchmarking and comparing speech recognition performance comprised of 6,659 radio transmission recordings from the KJFK local and ground controller positions. Within this set, 3,689 were pilot radio transmissions and 2,970 were local or ground controller radio transmissions.

The test data was recorded from the frequency record channels of the KJFK voice switch and was collected at the same time that the KJFK training data used during acoustic and language model training was collected. The recording circumstances ensure that the training data is a good representation of expected test data. All transmissions in the test data were set aside before training to eliminate model bias.

E. Experiment Methodology

Multiple experiment runs were performed on different combinations of acoustic and language models to identify superior speech recognition configurations. For each experiment run, the same acoustic model was used on all the test data, but the controller language model was used exclusively for the controller radio transmissions in the data, while the pilot language model was used exclusively for the pilot radio transmissions.

Word Error Rate (WER), which measures word-for-word omissions, additions, and substitutions in the recognized text, was used as a preliminary experimental measure to quickly pinpoint performance improvements across experiment configurations. WER is a reasonable measure of the general performance of a given speech recognition system/configuration, but it should not be used to infer application-specific performance. For this application, correct recognition of the ACID and command concepts is critical, but recognition of other words or phrases (such as wind information, traffic advisories, or courtesies) is not. Therefore, for low (i.e., better) WER benchmarks, the ACID pattern matching algorithm and the semantic meaning extraction algorithm were applied to the text hypotheses from the recognition engine to derive concept-level accuracy that was more relevant to the high-level application.

F. Recognition Performance Results and Findings

Initial WER benchmarks rapidly identified the adapted acoustic models and trained acoustic models as superior to the base acoustic model. Furthermore, as the quantity of training data used for training and adaptation expanded, the performance of the trained acoustic models began to surpass the performance of the adapted acoustic models, particularly on pilot speech. Table II summarizes a comparison of WER across the three different acoustic model types with a training data set of 127 hours of transcribed, silence-reduced audio.

TABLE II. COMPARISON OF WER ACROSS ACOUSTIC MODEL TYPES

	Generic AM	Adapted AM	Trained AM
Controller	29%	16%	15%
Pilot	61%	37%	32%

An additional observation the research team noted was the difference in rate of performance improvement as the quantity of training data expanded. In the case of acoustic model training, performance continually improved incrementally, suggesting the model could possibly improve further with more training data. In the case of acoustic model adaptation, performance improvement increased steadily at the beginning but also tapered, suggesting a diminishing rate of return.

Figure 3 illustrates the differences in rate of performance improvement between acoustic model types.

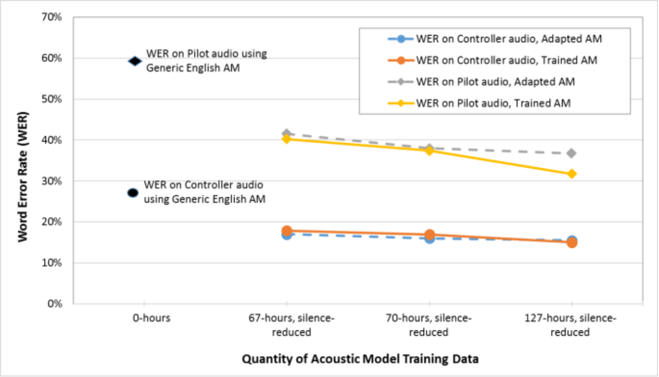


Figure 3. Comparison of performance improvement across acoustic model types.

Results from the custom acoustic model trained on 127 hours of silence-reduced audio were superior for both pilot and controller transmissions when compared to other configurations and were subsequently selected for further aircraft identifier and command concept analysis. Tables III and IV summarize the accuracy of aircraft identifier recognition and command concept recognition at the per-command level, where correct recognition of a command concept was defined as complete recognition of the command phrase and all associated parameter phrases and partial recognition was defined as recognition of the command phrase, recognition of one or more associated parameter phrases, or some combination thereof. Pilot results for the "Continue" command concept are not presented, as these command concepts did not appear in the pilot test set with sufficient frequency to generate representative metrics. In general, for both controller and pilot transmissions, the command concept recognition performance measures are much better than the preliminary WER measures. This is a promising finding because concept-level accuracy, which focuses on the recognition of intent-relevant words and phrases, is a better indication of future application-level alert performance.

Recognition accuracy on controller transmissions was notably higher than on pilot transmissions. In the case of controller transmissions, there were more instances of partial recognition than complete non-recognition of a command concept. Some of these cases could be addressed with the addition of context information, such as aircraft location and airport layout, in post-processing to correct errors in recognition. ACID recognition accuracy was notably lower than the accuracy observed on the command concepts. Analysis of the error cases indicated that in most instances the recognition engine recognized part of the ACID phrase correctly but final translation to the ACID symbol was thrown off by the erroneous recognition of one or two numeric words in the ACID. This error results because of the search space freedom enabled by the SLM and the wide variety of ACID phrases in the SLM training data. It can be mitigated by combining the ACID results of the SLM with results from a more restrictive language model such as a context-informed grammar, which has been shown to achieve correct ACID identification above 90%, or applying a more discriminant ACID extraction algorithm that identifies the closest, realistic ACID phrase, given the callsign and numerical words recognized [16].

Recognition accuracy on pilot transmissions was much lower than the accuracy observed on controller transmissions. It has significant room for further improvement. Analysis indicates that the system had greater difficulty here partly because pilot read backs often did not contain a command phrase and only contained the parameter(s) of the original controller command. As a result, the speech recognition system had fewer opportunities to find word patterns that were indicative of key content. Pilot responses also tended to be abrupt, choppy, and unpredictable, throwing off the system's pattern matching attempts. Possible options for improvement could be stronger keyword biasing based on preceding controller commands and additional acoustic model training to strengthen word identification despite background noise and speech disruptions.

TABLE III. ACID AND COMMAND CONCEPTS RECOGNITION ACCURACY ON CONTROLLER TRANSMISSIONS

	ACID	**LUAW**	**CFT**	**CTL**	**Hold Short**	**Cross**	**Turn Right**	**Turn Left**	**Taxi**	**Continue**
% Recognized Correctly	79%	94%	98%	91%	96%	96%	98%	97%	93%	97%
% Partially Recognized	NA	5%	1%	1%	3%	3%	2%	1%	2%	3%
% Not Recognized	21%	1%	2%	7%	1%	1%	0%	2%	5%	0%

TABLE IV. ACID AND COMMAND CONCEPTS RECOGNITION ACCURACY ON PILOT TRANSMISSIONS

	ACID	**LUAW**	**CFT**	**CTL**	**Hold Short**	**Cross**	**Turn Right**	**Turn Left**	**Taxi**
% Recognized Correctly	63%	86%	80%	84%	88%	92%	87%	84%	80%
% Partially Recognized	NA	7%	8%	3%	8%	4%	4%	5%	12%
% Not Recognized	37%	8%	12%	13%	4%	5%	9%	11%	7%

ACID recognition accuracy was also significantly lower than the accuracy observed on the command concepts, for similar reasons to those noted previously on the controller transmission ACIDs. However, in the case of pilot read back transmissions, dialogue context may help match pilot read back transmissions to controller command transmissions, despite low ACID recognition accuracy. A brief analysis of the dialogue structure between controller and pilot transmissions showed that in cases where a pilot transmission immediately followed a controller instruction or clearance, the pilot transmission was a response by the correct pilot over 99% of the time. Thus, it may be possible to use the timing of the transmissions to match pilot transmissions to controller transmissions, although this method is susceptible to stolen clearance errors.

G. *Case Study Example: Hold Short Instructions*

A read back error detection application must ultimately combine the results from both controller and pilot transmission recognition with alert generation logic (rules) to derive its final alert behavior. As noted Section III, defining appropriate alert generation rules can be nuanced and rules may vary across command types. Command-specific alert generation rules must still be developed and vetted with domain experts for a complete read back error detection application, but the rules must be defined with consideration of the entire application.

To illustrate the effects of alert generation rules on overall performance, and to get an early sense of application performance given the current preliminary speech recognition performance results, the research team simulated alert performance metrics for a single instruction—hold short of a runway—with a straightforward alert generation rule. This instruction was selected for the case study example because it is called out specifically in JO7110.65. The hold short instruction and read back recognition accuracy was assessed with the simple rule that an alert should be generated if any part of the instruction—the "hold short" phrase, the runway, or both—is omitted from the read back. The test does not include logic that compares ACIDs in the instruction and read back or logic that waits to determine if the controller will correct the read back error; the test only covers the speech recognition system's performance in determining if a read back of the instruction was complete and correct.

In the test set, there were a total of 220 runway hold short instructions issued by the local controller. Analysis of the manual ground truth transcriptions indicates that 199 of the 220 hold short instructions were read back completely in the corresponding pilot transmission. Of the remaining 21 hold short instructions, 12 were read back partially (either the hold short instruction or the runway was missing) and 9 elicited read backs that did not contain any mention of the hold short instruction or the runway. Many of these partial or missing read backs were immediately caught by the controller and corrected through a retransmission of the hold short clearance, but for the purposes of this simple test, these instances are counted as cases where an alert should be issued. In other words, for this preliminary evaluation of alert generation performance, the simple alert generation rule ignored multi-turn dialogue structure (i.e., dialogues that contained more than two transmissions), which contained subsequent corrective action and read backs.

According to the simple alert generation rule described earlier, the ground truth comparisons indicate 199 non-alerts cases and 21 alert cases. Analysis of the speech recognition results indicates that, on the 199 non-alert cases, 181 cases would have correctly not triggered an alert but 18 cases would have generated false alerts because of failure to correctly recognize part or all of the pilot read back. This result indicates room for improvement in recognition of pilot read backs of hold short instructions, which is expected given the 86% accuracy reported in Table III above.

In terms of expected alerts, analysis of the speech recognition results on the 21 alert cases indicates that 19 cases would have correctly triggered an alert—according to the simple alert logic rules defined—and 2 alert cases would have been missed because of incorrect recognition of the runway in the pilot's read back. This missed alert rate would improve with improved recognition performance on pilot speech, but this result also illustrates one way in which alert generation rules affect system alert performance: alert generation rules determine the number of expected alerts. The simple rule defined in this case study yielded approximately 10% of read backs for a particular instruction warranting an alert. Even with perfect speech recognition performance, this number seems too high to be operationally acceptable.

Alert rules that take into account multi-turn dialogues could further reduce the number of expected alerts, for example, by recognizing that the controller has corrected a missing read back, which was the case for 5 of the 9 expected alert cases where the read back did not mention the hold short instruction or the runway. However, more sophisticated alert generation rules may require better speech recognition performance. Another important consideration for alert generation rules is the ATC operation. Changing the alert generation rules to require only the instruction *or* the runway to be read back would reduce the number of expected alerts from this test by over half (12 of 21), but might not be an operationally acceptable rule; SME input and policy decisions must be considered.

The alert behavior simulated above demonstrates how speech recognition performance and alert generation logic combine to determine application performance. The results indicate that better recognition—particularly on pilot speech—is needed, but they also indicate that alert generation rules are critical to judging the speech recognition performance. In other words, although speech recognition performance can be measured in a vacuum, its suitability for a particular application cannot be fairly judged without the logic of the application also being in place. For application system performance to be fairly assessed, alert generation rules must be defined precisely, with consideration of speech recognition performance.

V. NEXT STEPS

Based on the findings of this feasibility study, MITRE is continuing to develop the read back error detection capability described in this paper. In the next year, the research team is planning to build out the application infrastructure that complements the speech recognition component, adding real-time context information that could better filter and parse meaning from the recognized text. The context ingestion and processing component will be part of a larger development effort to create and test an end-to-end prototype that accepts in real time both speech and context information and provides alerts to a graphical user interface. As a part of this effort, MITRE will continue to investigate the types of read backs that are acceptable to controllers, which may differ by command type, and will leverage subject matter expertise from ATC operational staff to create the command-specific alert generation logic of the application.

On the speech recognition side, the team expects to see performance improvements in recognition accuracy through the incorporation of additional training data from new airports and the use of advanced tuning techniques such as Deep Neural Network (DNN) training. The team will incorporate DNN training into the acoustic model training process, building on the models created during the feasibility study. The shift to DNN acoustic models is expected to yield a significant improvement in recognition accuracy [29]. Preliminary results from DNN training using the same training data yielded 20% improvement in controller WER and 10% improvement in pilot WER. Additionally, within an application framework, the added dialogue context of interleaved controller and pilot transmissions will enable dynamic language model adjustments that could yield further recognition improvements.

The findings of this research support the continued development of speech-based applications in the ATC domain. The rapid advancement of speech recognition technology and computer processing power in recent years has brought a number of envisioned future ideas within near-term reach. In our opinion, this transformative technology should be leveraged in the ATC domain, where voice communications are a pivotal presence in the day-to-day execution of safe and efficient operations.

Acknowledgment

The authors would like to thank Dr. Weiye Ma for her work in collating the audio and transcription training data used for this research.

References

[1] Kopald, H., Chen, S., Ma, W., & Britan, C. (2016). Analysis of Voice Source Options for Speech Recognition in Tower Operations (MTR160231). The MITRE Corporation. McLean, VA.

[2] Chen, S., Kopald, H. D., Chong, R.S., Goring, B. Levonian, Z., Wei, Y. & White, K. (2016). Methods for Expanding Speech Recognition Applications for Early Resolution of Surface Safety Events (MTR160300). The MITRE Corporation. McLean, VA.

[3] Federal Aviation Administration, Air Traffic Organization, *Air Traffic Control, Order JO7110.65W*, Washington, D.C.: U.S. Departument of Transportation, 2015.

[4] Cardosi, K. (1994). An Analysis of Tower (Local) Controller-Pilot Voice Communications. Federal Aviation Administration. Washington, D.C.: U.S. Department of Transportation.

[5] Cardosi, K. (1995). An Analysis of Ground Controller-Pilot Voice Communications. Federal Aviation Administration. Washington, D.C.: U.S. Department of Transportation.

[6] Cardosi, K. (1993). An analysis of En Route Controller-Pilot Voice Communications. Federal Aviation Administration. Washington, D.C.: U.S. Department of Transportation.

[7] Cardosi, K., Brett, B., & Han, S. (1996). An Analysis of TRACON (Terminal Radar Approach Control) Controller-Pilot Voice Communications. Federal Aviation Administration. Washington, D.C.: U.S. Department of Transportation.

[8] Cardosi, K., Falzarano, P., & Han, S. (1999). Pilot-Controller Communication Errors: An Analysis of Aviation Safety Reporting System (ASRS) Reports. Federal Aviation Administration. Washington, D.C.: U.S. Department of Transportation.

[9] Prinzo, O. V., Hendrix, A. M., & Hendrix, R. (2009). The Outcome of ATC Message Length and Complexity on En Route Pilot Readback Performance. Federal Aviation Administration. Washington, D.C.: U.S. Department of Transportation.

[10] G. van Es, "Air-ground communication safety study: an analysis of pilot-controller occurrences," European Organisation for the Safety of Air Navigation, Brussels, Belgium , 2004.

[11] Kopald, H., & Goring, B. (2015). Preliminary Shortfall Analysis for Applications of Speech Recognition for Surface Safety and the Closed Runway Operation Prevention Device (CROPD). McLean: The MITRE Corporation.

[12] Saon, G., & Chien, J.-T. (2012, November). Large-vocabulary continuous speech recognition systems: A look at some recent advances. IEEE Signal Processing Magazine, 29(6), 18-33.

[13] Beaufays, F., Bourlard, H., Franco, H., & Morgan, N. (2002). Speech recognition technology. In M. A. Arbib (Ed.), Handbook of Brain Theory and Neural Networks. Cambridge, MA: The MIT Press.

[14] Bahl, L. R., Brown, P. F., de Souza, P. V., & Mercer, R. L. (1993, January). Estimating hiddle Markov model parameters so as to maximize speech recognition accuracy. IEEE Transactions on Speech and Audio Processing, 1(1), 77-83.

[15] Baker, J., Deng, L., Glass, J., Khudanpur, S., Lee, C. H., Morgan, N., & O'Shaughnessy, D. (2009, May). Developments and directions in speech recognition and understanding, part 1. Signal Processing Magazine, 26(3), 75-80.

[16] Chen, S., Kopald, H. D., Elessawy, A., Levonian, Z., & Tarakan, R. M. (2015). Speech inputs to surface safety logic systems. IEEE/AIAA 34th Digital Avionics Systems Conference (DASC). Prague.

[17] Oualil, Y., Schulder, M., Helmke, H., Schmidt, A., & Klakow, D. (2015). Real-time integration of dynamic context information for improving autoamtic speech recognition. Interspeech. Dresden.

[18] Schmidt, A., Oualil, Y., Ohneiser, O., Kleinert, M., Schulder, M., Khan, A., & Helmke, H. (2014). Context-based recognition network adaptation for improving on-line asr in air traffic control. 2014 IEEE Spoken Language Technology Workshop (SLT 2014), (pp. 2-6).

[19] Shore, T., Faubel, F., Helmke, H., & Klakow, D. (2012). Knowledge-based word lattice rescoring in a dynamic context. Interspeech (pp. 1083-1086). Portland: 13th Annual Conference of the International Speech Communication Association.

[20] Gales, M. J., & Woodland, P. C. (1996, October). Mean and variance adaptation within the MLLR framework. Computer Speech & Language, 10(4), 249-264.

[21] Kuhn, R., Nguyen, P., Janqua, J. C., Goldwasser, L., Niedzielski, N., Finke, S., & Contolini, M. (1998). Eigenvoices for speaker adaptation. International Conference on Spoken Language Processing (ICSLP), (pp. 1771-1774). Kobe.

[22] Lee, C. H., & Gauvain, J. L. (1993). Speaker adaptation based on MAP estimation of HMM parameters. International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (pp. 558-561). IEEE.

[23] Leggetter, C. J., & Woodland, P. C. (1995, April). Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. Computer Speech & Language, 9(2), 171-185.

[24] Shore, T., Faubel, F., Helmke, H., & Klakow, D. (2012). Knowledge-based word lattice rescoring in a dynamic context. Interspeech (pp. 1083-1086). Portland: 13th Annual Conference of the International Speech Communication Association.

[25] Ragnarsdottir, M. D., Waage, H., & Hvannberg, E. T. (2003). Language technology in air traffic control. IEEE/AIAA 22nd Digital Avionics Systems Conference (DASC). Indianapolis.

[26] Huggins-Daines, D., Kumar, M., Chan, A., Black, A. W., Ravishankar, M., & Rudnicky, A. I. (2006). Pocketsphinx: a free, real-time continuous speech recognition system for hand-held devices. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Toulouse.

[27] Adapting the default acoustic model. (2016, 03 27). Retrieved from CMUSphinx: http://cmusphinx.sourceforge.net/wiki/tutorialadapt

[28] Training acoustic model for CMUSphinx. (2016, 10 14). Retrieved from CMUSphinx: http://cmusphinx.sourceforge.net/wiki/tutorialam

[29] Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., . . . Kingsbury, B. (2012, November). Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Processing Magazine, 29(6), 82-97.

Approved for Public Release; Distribution Unlimited. Case Number 17-0012.

This work was produced for the U.S. Government under Contract DTFAWA-10-C-00080 and is subject to Federal Aviation Administration Acquisition Management System Clause 3.5-13, Rights In Data-General, Alt. III and Alt. IV (Oct. 1996). The contents of this document reflect the views of the author and The MITRE Corporation and do not necessarily reflect the views of the Federal Aviation Administration (FAA) or the Department of Transportation (DOT). Neither the FAA nor the DOT makes any warranty or guarantee, expressed or implied, concerning the content or accuracy of these views.