Evaluation of speech engines for ATC Simulator

Priyanka Bhandia
Computer Science Engineering
PES University
Bengaluru, India
priyankbhandia@hotmail.com

Venkatarangan M.J
Electrical and Electronics Engineering
PES University
Bengaluru, India
venkataranganmj@pes.edu

Srivatsa S.K
Department of Basic Sciences
Atria Institute of Technology
Bengaluru, India
srivatsa.sk@pes.edu

***Abstract*— Many general speech engines are available that includes Automatic Speech Recognition (ASR) technologies and have been embraced for general applications. The present work is to examine the adoption of ASR technology in Air Traffic Control (ATC) simulation environment. The paper presents a survey and evaluation of these speech engines that can be integrated into ATC simulators. The ATC system is heavily dependent on verbal communication between the Air Traffic Controller and pilots which happens in varied phases between pre-flight to landing. The challenges that are normally faced to bring in the ASR technology to ATC domain are call sign detection, poor input signal quality, problems with ambiguity and use of non-standard phraseology. In this paper, we closely examine the performance of current ASR technologies based on different test scenarios and thus an assessment of current ASR technologies to apply to ATC is presented with quantitative figures and recommendations.**

Keywords*— *Air Traffic Control, ATC, Automatic Speech Recognition, ASR

I. INTRODUCTION

Many pilot and aviation trainings and tests use simulators heavily to develop understanding and to practice the ATC communication. It is to be noted that the ATC to pilot interaction is only verbal. Thus it is absolutely important to deploy an engine that is specific to ATC communication considering a limited and fixed vocabulary and sentences used in this domain. This is emphasised well with a good overview provided for ASR from ATC context, the general structure for recognition, modules and performance measurement in [1].

If the verbal communication protocol between ATC and pilots is understood well, it is very much possible to easily adopt ASR technology to ensure conformance to this limited prescribed syntax. When the ATC interacts with pilot in the simulation environment, there is a great scope to introduce validation of the pilot verbal responses using ASR (Automatic Speech Recognition) engines. Before the validation process, the important step is to recognise and decode the spoken words of the pilot trainer. The motivation is to make best use of this situation and introduce ASR technology into the ATC simulation environment to acquire the knowledge in a form suitable for validating the simulation engines.

The remainder of the paper is structured as follows: Section II gives an overview of system focussing on the scenarios in ATC communication and challenges. Section III provides an overview of the ASR engines, the technology behind and the parameters to be evaluated. Section IV provides comparisons and interpretations of results and recommendations for pilot training using simulators. Section V ends with conclusions and scope for further work.

II. ATC SYSTEM SCENARIOS

It is important to have a clear and unambiguous communication between pilots and ATC for safe and expeditious operation of aircraft navigation. There are standards set for this communication and hence it is important for pilots and ATC to use standard words and phrases to maintain a professional standard and more importantly to have error-free communication. ATC communication is an agreed standard procedure and process in aviation domain to ensure that the flights or aircrafts do not crash into each other. There are different controller roles namely Clearance and Delivery to check flight plans, Ground Control for movement of aircraft on ground, Tower Control for controlling active runways and clearing take-off and landing, Approach Control is for leaving or arriving at airport until they are in space and Centre control that owns air space not controlled by other controllers. Based on the live streams of communication available from various ATCs ([4] [5] and [6]), following are the main scenarios in flight that are to be considered for detailed analysis of voice communication exchange:

1) Pre-flight: This portion of the flight starts on the ground and includes flight checks, push-back from the gate and taxi to the runway. Pushback is an airport procedure during which aircraft is pushed backwards away from airport gates by external power like low profile vehicles called pushback tractors or tugs. This includes communication of when the flight is ready for pushback, requesting for taxi for pushback and completing the procedure.
2) Take-off: Pilot powers up the aircraft and speeds down the runway. The communication includes final clearance for take-off when it is deemed safe by the controller.
3) Departure: Plane lifts off the ground and climbs to a cruising altitude. The communication involves the instructions to pilot about heading direction, speed, rate of ascent.
4) En-route: Aircraft travels through one or more centre airspaces and nears the destination airport. The communication involves updates to pilot about weather and air traffic information. It also includes instructions to pilot on speed and altitude to maintain separation between aircrafts within a sector.
5) Descent: Pilot descends and manoeuvres the aircraft to destination airport.
6) Approach: Pilot aligns the aircraft with designated landing runway. Here, communication involves the directions to pilot to adjust heading direction, speed and altitude to line up to prepare for landing.
7) Landing: Aircraft lands on the designated runway, taxis to the destination gate and parks at the terminal. The communication involves the local controller to inform pilots about the clearance for landing.

III. ASR ENGINES

The ASR engines available at this point of time that could be considered deployable for ATC domain are briefed below:

A. *Sphinx4*

Sphinx4 is a continuous-speech, speaker-independent recognition system making use of hidden Markov acoustic models (HMMs) and an n-gram statistical language model. It can be used with Java and accepts W3C compliant grammars. It works offline

B. *PocketSphinx*

It is a version of Sphinx written in C that can be used in embedded systems.

C. *Julius*

It is an ASR engine written in C whose English model is unavailable for redistribution. It also works offline.

D. *.NET System.Speech*

It is an ASR engine available as a part of Microsoft's .NET Framework and can be used in C# and Visual Basic programs. It can consume grammars which are W3C compliant and is available as part of the Windows OS since Windows Vista. It works offline.

Among the ASR engines described above, only .NET System.Speech and Sphinx4 are chosen for further evaluation. The Julius speech engine is not redistributable, and PocketSphinx is a sibling of Sphinx4 written for use in mobile devices and embedded systems and hence not considered for further evaluation. The technology behind ASR, not dealt in this paper, has been reviewed thoroughly in [9] and [10]. The increased performances from ASR and hence an increased acceptance based on the data is discussed in [11].

IV. Evaluation Criteria and Results

A. *Inputs to the Simulation*

The experiments were run on a collection of twenty audio recordings of ATC communications.

Each audio recording contains the interaction between pilots and the air traffic controller during the different scenarios as referred in Section II. These audio recordings are named benchmark1, benchmark2 to benchmark20 and were taken from [4] that provide live and recorded ATC audio transmissions. The engines will have a dependency on factors primarily on dialects of the pilots as it depends on their nationality and also the background noise that would exist within cockpit. Table 1 and Table 2 lists the representative test cases with different noise characteristics that were selected for evaluation.

The test setup is as described in [7]. Tests were carried out on consumer grade hardware on the Windows 7 operating system. For the ASR engines tested, no acoustic model training was done. For the ASR engines that required the specification of a grammar, all the words used in all the audio recordings were specified but sentence structure was not specified. There have been references to on how comparisons of ASR engines can be done quantitatively in [3] and [7] but only part of it has been taken into account as the focus was not on multiple dialects yet.

TABLE I. Audio File Contents

Audio Recording File Name	Phrases Spoken in the Recording
benchmark1	at your eleven o clock at your one o clock and four miles north bound is a triple seven heavy load
benchmark2	lear jet four three niner delta cleared approach runway seven
benchmark3	one four charlie turn left heading two seven zero
benchmark4	one fourteen charlie descend maintain niner thousand
benchmark5	lear jet seven two golf fly direct planes climb maintain flight level two zero zero
benchmark6	one one four charlie cleared approach runway three five right
benchmark7	november six four zero charlie contact center
benchmark8	eight one eight delta contact tower
benchmark9	seven seven two golf contact denver center
benchmark10	united six seven roger ready for descent
benchmark11	united six seven resuming navigation direct contact channel approach one two one decimal four
benchmark12	negative no comms on uniform roger
benchmark13	car nine eight whiskey and company runway two two eight proceed via quebec hold short of November
benchmark14	car nine eight whiskey continue left november left alpha hold short of victor
benchmark15	alpha bravo delta six six six heavy
benchmark16	leer jet three two zero golf turn left heading zero nine zero
benchmark17	three twenty golf descend maintain one one thousand
benchmark18	november six four zero charlie fly direct rocky climb maintain flight level two three zero
benchmark19	three two zero golf contact tower
benchmark20	november eight one eight delta turn right heading one six zero descend maintain ten thousand

TABLE II. Noise Characterstic Of Audio Recordings

Audio Recording File Name	Noise Characteristic
benchmark1, benchmark8, benchmark9, benchmark11	Considerable background noise
benchmark7, benchmark14, benchmark15	Some Background noise
benchmark2, benchmark3, benchmark4, benchmark5, benchmark6, benchmark12, benchmark13, benchmark16, benchmark17, benchmark18, benchmark19, benchmark20	No background noise

B. *Accuracy*

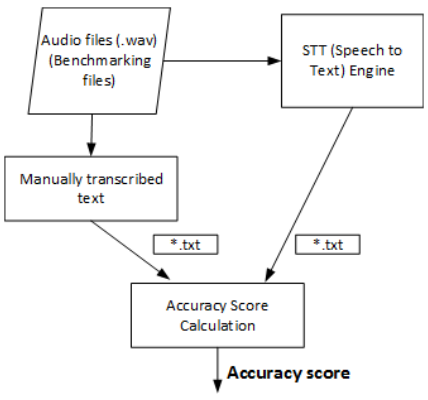

Fig. 1. Flow chart showing the approach to calculate the accurcy score

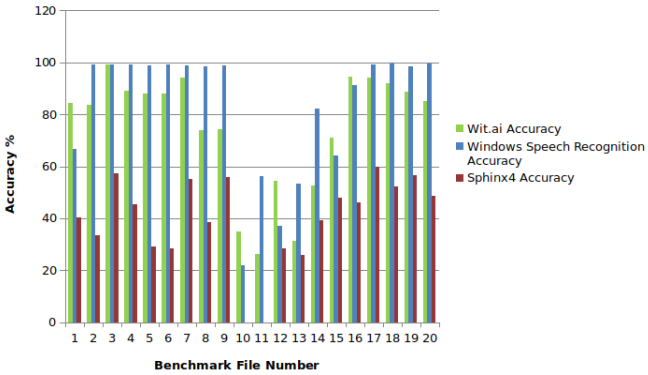

Fig. 2. Graph of accuracy % of ASR engines in which wit.ai speech recognition service is used as a benchmark.

The accuracy of the ASR Engines is measured against manual transcriptions of the audio files. The text output of the ASR Engine for an audio file and the manual transcription of the same audio file are given as input to the Accuracy Score Calculation Script, as show in Fig 1. The script makes use of the Gestalt Pattern Matching Algorithm [12] to measure the accuracy, treating the manual transcription as the source of truth. Punctuations and case are ignored while calculating the accuracy of the ASR Engine. Fig 2 shows accuracy results using online speech recognition service Wit.ai as a benchmark.

TABLE III. ACCURACY OF ASR ENGINES

Audio Recording File Name	**Accuracy % of ASR Engine**		
	Wit.ai	**Windows Speech Recognition**	**Sphinx4**
benchmark1	84.375	66.666	40.322
benchmark2	83.606	99.186	33.333
benchmark3	99.047	99.065	57.142
benchmark4	89.108	99.047	45.454
benchmark5	88	98.888	29.23
benchmark6	88.135	99.186	28.571
benchmark7	94.117	98.901	55.263
benchmark8	73.846	98.591	38.461
benchmark9	74.418	98.823	55.696
benchmark10	34.92	21.738	0
benchmark11	26.143	56.296	0
benchmark12	54.545	37.209	28.571
benchmark13	31.325	53.424	25.714
benchmark14	52.459	82.089	39.344
benchmark15	70.967	64.15	47.826
benchmark16	94.308	91.056	46.153
benchmark17	94.117	99.029	59.74
benchmark18	91.954	99.447	52.307
benchmark19	88.524	98.507	56.521
benchmark20	85.082	99.459	48.648

The observed accuracy values were also expressed as a percentage of the accuracy of the benchmark ASR Engine, the values are presented in the graph shown in Fig 3 and Table IV respectively.

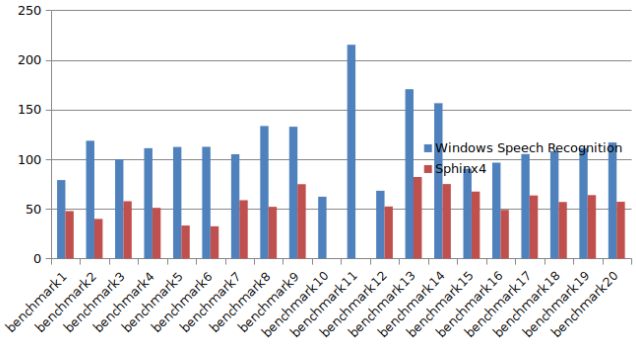

Fig. 3. Graph of accuracy scores for ASR engines for different audio recordings showing out performance of Windows Speech Recognition of all test cases when compared to Sphinx4

TABLE IV. ACCURACY OF ASR ENGIE AS A PERCENTAGE OF BENCHMARK ENGINE ACCURACY

Audio Recording File Name	**Accuracy as % of Benchmark Accuracy**	
	Windows Speech Recognition	***Sphinx4***
benchmark1	79.0115555555556	47.789037037037
benchmark2	118.635026194292	39.8691481472622
benchmark3	100.018173190506	57.6918028814603
benchmark4	111.153880684114	51.01001032455
benchmark5	112.372727272727	33.2159090909091
benchmark6	112.538719010609	32.4173143473081
benchmark7	105.083034945865	58.7173411817206
benchmark8	133.508923976925	52.0827126723181
benchmark9	132.794485205192	74.8421080921282
benchmark12	68.2170684755706	52.3806031716931
benchmark13	170.54748603352	82.0877893056664
benchmark14	156.482205150689	74.9995234373511

benchmark15	90.3941268476898	67.3918863697211
benchmark16	96.551724137931	48.9385842134283
benchmark17	105.219035880872	63.4741863850314
benchmark18	108.14863953716	56.8838767209692
benchmark19	111.277167773711	63.8482219511093
benchmark20	116.897816224348	57.177781434381

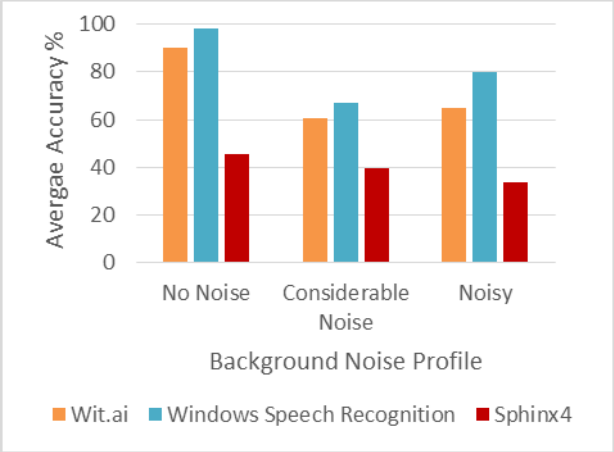

Fig. 4. Outperformance of Windows Speech Recognition across different background noise profiles, as per the classification of files in Table 2

C. *Recognition Time*

The time taken to complete the speech recognition task was measured for the ASR engines and the comparison is shown in Fig 5 and Table V. The times measured do not include the time taken to read the files as that reflects performance of physical storage system and not of the engines.

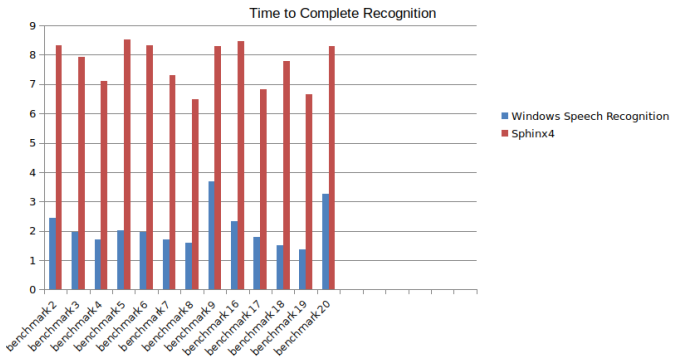

Fig. 5. Graph of time taken to complete Speech Recognition task for ASR engines. (It is noticed that WSR takes less time and is approximately 3.7 times faster compared to Sphinx4)

TABLE V. TIMES TAKEN IN SECONDS BY ASR ENGINE TO COMPLETE SPEECH RECOGNITION TASK

Audio Recording File Name	**Time in seconds to Complete Recognition**	
	Windows Speech Recognition	***Sphinx4***
benchmark2	2.447	8.32
benchmark3	1.965	7.922
benchmark4	1.699	7.1
benchmark5	2.014	8.527
benchmark6	1.953	8.31
benchmark7	1.716	7.295
benchmark8	1.592	6.497
benchmark9	3.674	8.282
benchmark16	2.316	8.465
benchmark17	1.788	6.812
benchmark18	1.505	7.785
benchmark19	1.378	6.658
benchmark20	3.266	8.282

D. *State handling*

Every phase of flight is modelled as a state, where each state consists of verbal exchanges that take place between a pilot and the air traffic controller. These states are placed in a linked list which models the communication that takes place between a pilot and air traffic control for a full flight as a series of states. This model was implemented in C# using the System.Speech ASR Engine and the air traffic controller replies were vocalized using the System.Speech speech synthesizer. The ASR Engine was set to recognize the English language. The microphone used did not have any noise cancelling features. This model was tested to find the confusion matrix. The speaker in this test was a nineteen year old Indian male. The average background noise for this test was 44.7 dB. The distance of the speaker from the microphone was about half a meter. The length of the messages spoken varied from five to twenty four words. The results are shown in Table VI and Table VII.

TABLE VI. CONFUSION MATRIX OF MODEL WITHOUT NOISE CANCELLING MICROPHONE

Model performance without noise cancelling microphone		
	Accepted as Correct	**Rejected as Incorrect**
Correct Message	129	21
Incorrect Message	83	67

TABLE VII. CONFUSION MATRIX OF MODEL WITH NOISE CANCELLING MICROPHONE

Model performance with noise cancelling microphone		
	Accepted as Correct	**Rejected as Incorrect**
Correct Message	145	5
Incorrect Message	66	84

A total of three hundred messages were given as input to the model of which one hundred and fifty were correctly spoken. In the other half, errors in the call sign of the aircraft and incorrect information about the flight plan were introduced to check for false positives.

This test was repeated with a noise cancelling microphone which wraps around the speaker's head. With the same number of incorrect and correct messages as input.

Here, the accuracies shown both with and without the external noise cancelling microphone are around 70%. Sensitivity is also comparable with and without external microphone, at 65.8% and 63.4% respectively. However, there is a drastic difference in the specificities and precisions, with the model without the microphone having a specificity of 79.8% and the model with the microphone having specificity of 93%. The respective precisions are 86% and 96.6%.

These results show that the use of a noise cancelling microphone to provide input to the ASR Engine helps to reduce the occurrence of false negatives and positives. Higher precision obtained by use of noise cancelling microphone can be useful when the input can be viewed as having unbalanced classes, that is, more correct inputs than incorrect ones, as fewer correct inputs will be rejected as incorrect, thus reducing unnecessary repetition by the pilot.

TABLE VIII. DIFFERENT MODEL PARAMATERS

Different Model Parameters in %		
Parameter	**With Microphone**	**Without Microphone**
Accuracy	70.4	70.6
Sensitivity	63.4	65.8
Specificity	93	79.8
Precision	96.6	86

V. CONCLUSIONS

In this paper, the ASR engines available in market, some open source and a few licensed are listed. Only the ones that were suitable for further evaluation were taken and tests were conducted. The paper presents the test data used that represent the scenarios of ATC communication along with the test results quantitatively using accuracy, benchmarking figures, confusion matrix, sensitivity, specificity and precision. Windows Speech Recognition is clearly on the lead considering the applicability of it to ATC domain with results proving the same. The test data selection has been taken from online live data stream so that it is more realistic. The work also included the state handling as part of the test routines with consideration of the expected voice messages for each state.

As we expect more technical details regarding the ASR engines to be made available, a thorough qualitative analysis including other engines will be taken up in the following study. Further, a similar study is planned for other contexts (like industry specific ASR requirements that can be included in the industry 4.0 – industrial automation and for Industrial IOT applications.

ACKNOWLEDGEMENT

We thank AERX labs, Bengaluru (India) for giving us an opportunity to work on this problem statement and supporting with on-campus simulator infrastructure and review support. We also recognise the support and work of first year students namely Mr. Alok Mehendale, Mr. Arpit Agarwal and Mr. Parth Shah by providing state handling and testing.

REFERENCES

[1] Nguyen, V. N., & Holone, H. (2015). Possibilities, challenges and the state of the art of automatic speech recognition in air traffic control. World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering, 9(8), 1940-1949

[2] Kopald, H. D., Chanen, A., Chen, S., Smith, E. C., & Tarakan, R. M. (2013, October). Applying automatic speech recognition technology to air traffic management. In Digital Avionics Systems Conference (DASC), 2013 IEEE/AIAA 32nd (pp. 6C3-1). IEEE.

[3] Shen, W., & Reynolds, D. (2007). A comparison of speaker clustering and speech recognition techniques for air situational awareness. In Eighth Annual Conference of the International Speech Communication Association

[4] "Live ATC streams," [Online]. www.liveatc.net

[5] "Flight Communications" [Online] https://www.firstflight.com/private-pilot-course/flight-communications/

[6] ATC communication http://atccommunication.com/

[7] Speech Recognition Experiments with the TRACON/Pro ATC Simulator. EuroControl Experimental Center

[8] Geacăr, C. M. (2010). Reducing pilot/ATC communication errors using voice recognition. In Proceedings of ICAS (Vol. 2010).

[9] Ghai, W., & Singh, N. (2012). Literature review on automatic speech recognition. International Journal of Computer Applications, 41(8).

[10] Radha, V., & Vimala, C. (2012). A review on speech recognition challenges and approaches. doaj. org, 2(1), 1-7.

[11] Helmke, H., Ehr, H., Kleinert, M., Faubel, F., & Klakow, D. (2013, June). Increased acceptance of controller assistance by automatic speech recognition. In Tenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2013) (pp. 1-10).

[12] "Pattern Matching: The Gestalt Approach" [Online] http://collaboration.cmc.ec.gc.ca/science/rpn/biblio/ddj/Website/articles/DDJ/1988/8807/8807c/8807c.htm