Voice Controlled Augmented Reality: A Comparison of Speech-Recognition Tools for AR Applications

Juan Estrella and Benito Mendoza
New York City College of Technology

Abstract

Augmented Reality (AR) refers to the technologies that enhance the version of the physical environment with computer-generated sensory input such as sound and graphics overlaid on top of the user's view of the real world. Artificial Intelligence (AI) studies how to make computer programs and machines "smart." Our research project focuses on exploring the Integration of AI in AR applications. Specifically, on using Speech Recognition or Natural Language Processing for controlling virtual AR objects and enhancing the human-computer interaction. It is obvious that integration of AI and AR is of great value. However, for developers, it is difficult to find the right tools to start building applications. We present an empirical study that compares currently available alternatives for creating voice-controlled systems. We compare several Speech Recognition services in terms of openness, usability, cost. We developed two applications to test these services, one that uses simple keyword-based voice commands and the second that uses more advances sentences. We present our experience while integrating these libraries/services with the game engine used to develop AR applications, and the services pros and cons.

Keywords

Artificial Intelligence, Augmented Reality, Speech Recognition, Natural Language Processing.

Introduction

Augmented Reality (AR) and **Artificial Intelligence (AI)** are two of the most important frontiers of technological development. However, as it stands, these two fields are not as integrated as is believed. Although, with future research and development, the potential for more effective integration is attainable.

AR is a technology that superimposes digital content, generated by computers, over real-world environments; it provides an enhanced view of the real world. AR provides an aid in visualization by simulating 3D objects that can be manipulated. For example, as shown in Fig.1, engineers can view their 3D virtual designs and modify them before the final physical production [1]. With AR it is also possible to simulate phenomena that are invisible to the naked eye, such as wind, electrical current in a circuit, magnetic fields, or gravity[2]. Furthermore, AR can portray features of large real-world facilities on a smaller scale.

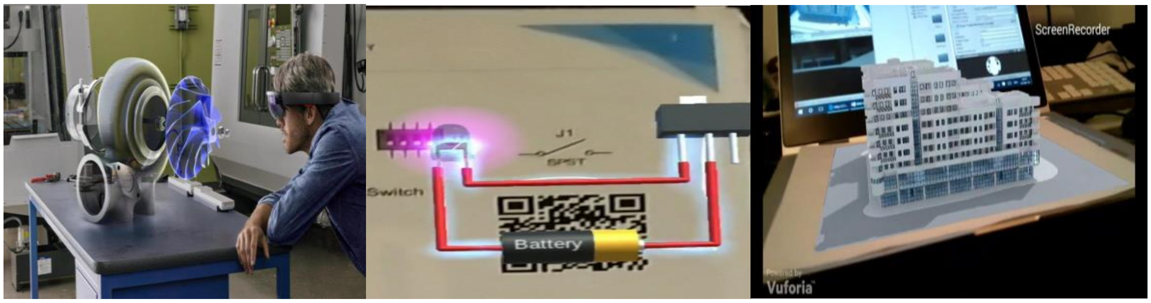


Figure 1. AR Visualization advantages: (a) Finding Flaws before production, (b) Simulating phenomena such as electric flow in a circuit, and (c) displaying large scale features in confined spaces.

AI aims to make computer programs or machines to learn and take decisions. The most accepted approach to AI, the so-called Narrowed AI, uses algorithms and statistical models to learn from data sets to classify things and to make predictions, this is known as **Machine Learning (ML)**. A subfield of AI is **Computer Vision (CV)**, which deals with the capabilities of machines to interpret and understand the visual world from images on videos. Modern CV techniques are based on Deep Learning **(DL)**, a modern approach to ML based on Neural Networks, to accurately identify and classify objects. **Natural Language Processing** (NLP), one of the primary branches of AI, focuses on making computers able to understand the meanings of uttered sentences of a conversation, including context. **Speech Recognition**, such as the ones used on Amazon's Alexa and Google use DL and some of the NLP methods to allow computers to receive spoken inputs and recognize sentences.

AI and AR seem ideally suited to one another. In fact, AR relies on AI to be effective. Computer vision is used on AR for recognition of gestures, such as eye tracking, hand gestures, and others to control applications. However, there is no standardized approach for developing AR applications that involve other AI techniques such as Machine Learning and Speech Recognition. Developers must explore different alternative to integrate the Speech Recognition and AI. Developers find it difficult to integrate AI with AR because there is a lack of proper best practices, resources, and tools to develop their projects. There are many ways to combine AI with AR but, developers must find effective ways to combine tools and approaches to craft their projects. For example, there many library options for Speech Recognition that can potentially be integrated with AR. However, it is not clear which one integrates better with the AR development platforms.

In this paper we present a comparison of five Speech Recognition (SR) services, in terms of cost, openness and compatibility with a game engine to develop AR applications. We developed two applications that integrate AI and AR at different levels. The first application involves reacting to uttered phrases or keywords only. The second application consists reactions triggered by complete sentences. We believe that our comparison can be of help for AR developers looking to integrate Speech Recognition into their apps.

The rest of the paper is organized as follows: In Section II presents a brief history of Speech Recognition technologies, focusing integrating SR and AR. In Section III, we present the methodology use to develop the two applications mentioned above. Furthermore, we discuss how these applications were used to compare the SR services. In Section IV, we present the results of our comparison. Finally, in Section V, we present a short summary of our findings based on our experience developing these two applications and their compatibility with SR and we project next steps for future research.

Background

Speech Recognition **(SR)** was invented and introduced during the 1950s. For years, the process was time-consuming because most SR systems were only able to understand a limited number of words which were translated into digits rather than text. And even with developed capacities over the years, efficiency was lacking. Despite the large vocabularies of past SR systems, like Kurzweil's and IBM's, these programs received "discrete dictation," which required the speaker to pause after every word[3]. Furthermore, past SR systems were slow to respond to user inquiries. Although, of course, these aspects have improved since. Today, when using Speech Recognition software, like Alexa or Siri, users address the products by name to command them to listen for utterances or sentences, which serves as requests that will yield desired results. Speech Recognition technology has developed immeasurably, from the primitive stage of acknowledging single syllables, to constructing a vocabulary of thousands of phrases, and finally to answering questions with fast, witty answers, like the smart virtual assistant, Siri[4].

Manufacturing industries are interested in the benefits from the integration of Speech Recognition and Augmented Reality. The potential of AR for applications such as equipment maintenance, inspection, and modeling are the few that have been explored. For example, Aouam et.al.[5] describes an application for the automobile assembly/disassembly industry. The system recognizes voice commands that are translated into textual commands that are interpreted by an AR system. The AR system then allows the user to manipulate the different parts and view them by simply giving voice commands, such as the designation of objects by their name to make them appear or move them indicating the next position[5]. One more example, aerospace maintenance is time consuming and complex for most technicians, but when they are occupied with their hands, SR can provide a timesaving means for completing the task. The integration of SR and AR allow an Aviation mechanic to check the torque values of a Skylight while performing assembly to stay on track and be aware of completion as he is working[6]. Aviation mechanics can use these technologies as tools for 3D modeling their equipment and detect flaws for a better result. Industrial workers now use these technologies to their advantage to increase work performance and provide a great deal for training without compromising throughput and quality where time is needed the most[6].

In the literature there are some comparisons of speech recognition engines and their benefits[7]. The most popular engines are Google Cloud Speech API, IBM Watson Speech to Text, and Microsoft Azure Speech API. Each engine is similar in terms of cost because the more they are used the higher the cost is overtime and they have similar interfaces. However, none of the comparison address the integration of Speech Recognition with AR. In this paper, in addition to include other more recent developed engines, we focus in the integration of these speech recognition engines with AR.

Methodology

To test the capabilities of these SR engines we explore two types of applications. Both concentrate on manipulating virtual objects in an AR app; however, each follows a different approach. The first one operates with simple commands based on keyboards. The second operates based on understanding the context from commands in sentences. The First app involves a 3D figure of a Barbarian Warrior that performs certain animations when given specific voice commands. The Barbarian starts in an idle state and waits for commands to execute. Commands such as “Round Kick”, “Run”, and “Walk”, are simple keywords that trigger specific animations. I’m comparing these applications on how well they respond with each speech engine.

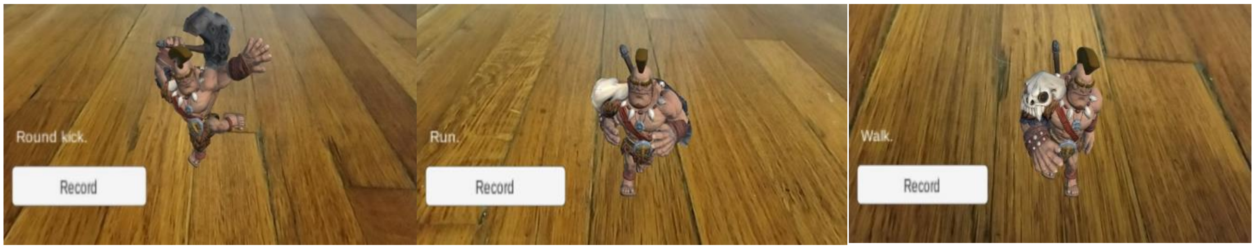
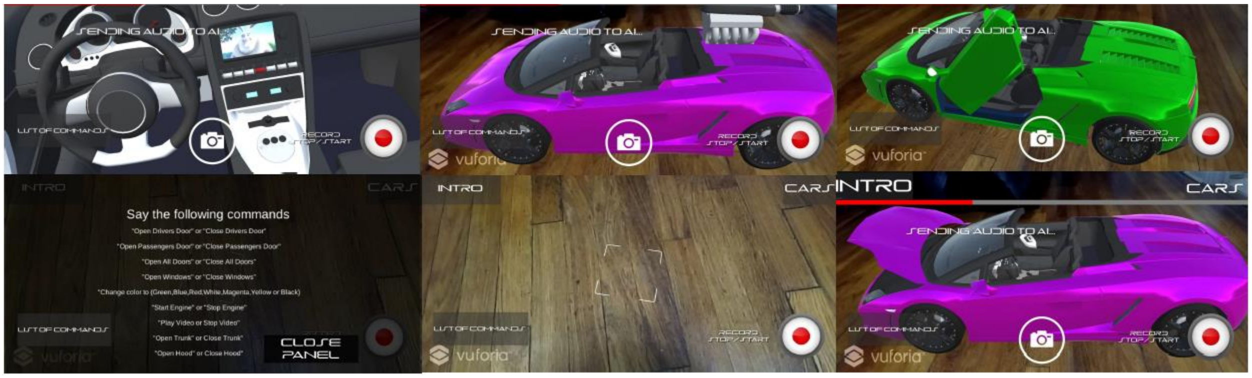


Figure 2. Controlling the Barbarian Warrior with Commands (a) “Round Kick” (b) “Run” and (c) “Walk”.

The second app involves commands consisting of more complex sentences, not just simple keywords. In this application we control a virtual AR car show room that allows us to spawn 3D models of cars and control them with specific commands using voice. We can bring a large-scale object at a smaller scale in the real world and interact with it using a mobile phone or tablet. The type of commands we can use are “Open the Driver’s Door”, “Close the Driver’s Door”. “Open the Hood”, “Close the Hood”, “Change color to [Specified Color]”, etc.



Say the following commands

"Open Drivers Door" or "Close Drivers Door"

"Open Passengers Door" or "Close Passengers Door"

"Open All Doors" or "Close All Doors"

"Open Windows" or "Close Windows"

"Change color to (Green,Blue,Red,White,Magenta,Yellow or Black)

"Start Engine" or "Stop Engine"

"Play Video or Stop Video"

"Open Trunk" or Close Trunk"

"Open Hood" or Close Hood"

Figure 3. Controlling a virtual automobile using complex sentences.

The applications were developed on the Unity game engine in combination Vuforia (an Augmented Reality platform that is integrated with Unity. We tried five different Speech Recognition services using their corresponding APIs: Google’s Cloud Speech-To-Text (STT), Microsoft’s Azure Speech-To-Text API, IBM Watson’s Speech-To-Text, Mozilla’s DeepSpeech, and Wit.ai. The main features of each one of these services are listed in Table 1, a description is presented next.

Table 1: Table of comparison of each Speech Recognition Service explored.

	Google's Cloud Speech-To-Text	Microsoft Azure Speech-To-Text	IBM Watson Speech-To-Text	Mozilla DeepSpeech	Wit.ai
Pricing	0.006 USD per 15 seconds.	1 USD per Hour	0.03 USD per minute	Free	Free
Multiplatform	Windows & Mac OS	Windows & Android	Windows, Android, Linux, Mac OS & IOS	Windows & Mac OS	Android, IOS, Mac OS & Windows
Integrate with Unity	Yes	Yes	N/A	N/A	Yes

Google's Cloud Speech-To-Text (STT) is a free API for the first 60 minutes, available with your Google account. If you sign up with a Credit/ Debit Card, Google provides a 12-month free account with $300 to use in any of their cloud services, including Speech-To-Text, Text-to-Speech, Computer Vision, and much more. Google's Speech-To-Text is supported on many platforms, Windows, OSX, IOS, Linux, Ubuntu and Android. It includes over 120 languages in its database.

Microsoft's Azure Speech-To-Text API is free for 12 months for students with a Microsoft account including $200 for the first three months since creating the account. Like Google's Cloud service people can use the $200 in any of their cloud services, including Speech-To-Text, Text-to-Speech, Computer Vision, and much more. This service provides the ability to "Pay as you go/use" or on a monthly/yearly subscription.

IBM Watson's Speech-To-Text is free for 12-months for everyone and is a subscription-based service. The service is priced at $0.03 (USD) per minute. This price applies to use of both broadband and narrowband models. Although it contains a powerful real-time Speech Recognition engine the drawback is that it can only transcribe audio from 10 languages, Arabic, English, Spanish, French, Brazilian Portuguese, Japanese, Korean, German, and Mandarin.

Mozilla's DeepSpeech is an open source Speech Recognition engine using structured models of data trained by Machine Learning techniques and uses Google's TensorFlow engine to make implementation easier. DeepSpeech uses Python, as coding language, to develop and train models, in this case to enhance its speech recognition service. DeepSpeech is supported on many platforms ranging from Windows to OSX, Linux, Raspbian and even Android devices like the Google Pixel 2. In addition, DeepSpeech supports many languages and improves the more language models are created.

Wit.ai is a free and straightforward cloud service owned by Facebook. This cloud-based platform allows developers to create a bot to recognize phrases and commands. Then this bot can be used in different applications. It follows a learning cycle where the user states some command sentences, the bot is trained with expected answer, the bot then learns and performs a response based on validation. The bot takes the user text or voice input and returns intents and entities. An intent is simply what the user intends to do. This could be something like *changeTemperature* or *getNews*. Entities are variables that contain details of the user's task. With an intent like *changeTemperature*, the user would want to be able to extract the exact temperature the user is trying to change it to from their text/voice input. Wit.ai comes with built-in entity types like *location*, *number*, and *amount_of_money*, plus you can create your own. Depending on the intent and entities obtained from the user input, an application can take actions or ask more questions to fulfill the user's request.

Results

The integration of Unity with directly with Google Speech-To-Text API failed. Unity lacks the proper libraries to effectively integrate with Google's STT; some Unity's libraries are deprecated to work with the current version of the API. However, Unity provides an asset where people can purchase an outdated version of Google's Voice Recognition for $20 but it is not completely accurate on recognizing phrases and sentences which results to displaying misunderstood context. Unity relies on special SDK's (Software Development Kits) to be compatible with Google's Speech-To-Text API for purposes of Speech-Recognition.

We explored the integration of Azure's Speech-To-Text API with Unity. We were able to control the Barbarian Warrior's behavior with Speech Recognition. After pressing the UI button on the screen of the app and under 15 seconds, we uttered commands such as "Run", "RoundKick" and "Walk". The audio would be sent to Microsoft's servers to be analyzed and sent back to display the uttered phenomes. The application is compatible with Unity, but Azure Speech-To-Text API currently supports Windows and Android devices. This application is like the app we developed using Wit.ai because it required certain scripts combined with phrases and keywords, activation keys and internet access in order to control the Barbarian Warrior.

The integration of IBM's Watson Speech API with Unity failure because as of now the SDK needed to work with Unity was deprecated back in early 2018, preventing an integration with Unity. There are studies that have shown the strong capabilities of IBM's Watson Speech API, particularly in Speech Recognition. Watson's Speech-To-Text is multiplatform and accurate on understanding uttered sentences from users and transcribing them. If the service was compatible with Unity, developers could potentially integrate its Speech Recognition engine with Augmented Reality and control virtual objects easily because of its accuracy.

We explored Mozilla's DeepSpeech service on Windows and realized that the software's interface is complex because in order to use DeepSpeech third party software must be installed. For example, we had to install Python 3 in order to extract the information from pre-trained English models for performing speech-to-text. DeepSpeech does not require internet access to recognize uttered phrases from users because it takes a local recorded .wav audio file, converts the audio and displays the phrases into text. Also, this service is accurate most of the time and compatible with Windows and MacOS. However, in order to integrate DeepSpeech with Unity, we must use third party plugins that supports the Python Language which does not help us because we are searching for an easy to use Speech Recognition software that can be integrated with Unity in C# language.

The combination of Wit.ai, Unity, and Vuforia is accessible to everyone at zero cost and works very well. Developing the applications of controlling a Barbarian Warrior and an AR virtual car with Unity and Wit.ai is like the Azure Speech-To-Text API application. It involves an account, an API key, and scripts calling the Wit.ai API from a server. Also, the conversion of audio is similar, .wav audio file is sent to the server, to be process by the Speech Recognition service. The resulting text is sent back to the application. However, Wit.ai requires internet access in order to perform the commands. Aside from that issue, Wit.ai meets our demands because it's simple, reliable, compatible with Unity and improves its resources every day.

Conclusions

From our empirical comparison of Speech-Recognition services, we concluded that Wit.ai takes the win. Wit.ai has a simple user interface, completely cost-free service, and the integration with Unity is simple. Each speech engine had similar integration tactics, consisting on calls to functions from a web-based API. However, Google and IBM's Speech-To-Text were hard to integrate because they were not entirely supported within Unity's libraries. DeepSpeech is still new; it works with Python 3. Unity's scripting language is C#, so the integration is not trivial. DeepSpeech libraries and Python 3 must be installed in the device that will use them. Microsoft's Azure Speech-To-Text API worked well, even with the trial version our apps work well. However, the trial version is only supported on Windows and Android devices. Azure's full version might be better because it has no limitations.

We believe that voice recognition within an AR environment can greatly influence the human-machine interaction by allowing it to communicate, control, and interact with virtual and real objects more naturally, especially in situation when the user is using his/her hands for other tasks. As for future work, we are exploring the integration Speech Recognition with Microsoft's HoloLens to control non-virtual objects, such as remote-controlled car.

References

1. Zhang, H. H. H. (2018) "Virtual Prototyping for Mechatronics." Proceedings of the 2018 ASEE Mid-Atlantic Section Spring Conference, Washington, District of Columbia. https://peer.asee.org/29501
2. Gargalakos, M., Rogalas, D. (2012) "The EXPLOAR project: Visualizing the invisible." Proceedings of the Proceedings of Science Center to Go Workshop in Augmented Reality in Education. Athens, Greece, October 27 - 29, 2011.
3. Huang, X., Baker, J., and Reddy, R. (2014) A historical perspective of speech recognition. Communications of the ACM. 57, 1 (January 2014), 94-103. DOI: https://doi.org/10.1145/2500887
4. Pinola, M. (2011) "Speech Recognition Through the Decades: How We Ended Up With Siri". PC World. November, 2011. Retrieved 24 October, 2019.
5. Aouam, D., Benbelkacem, S., Zenati, N., Zakaria, S., Meftah, Z. (2018) "Voice-based Augmented Reality Interactive System for Car's Components Assembly." Proceedings of the 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS). Tebessa, Algeria. 24-25 Oct. 2018
6. Ballard, B. (2018) "Giving voice to augmented reality." Aerospace Manufacturing and Design. July 3, 2018. https://www.aerospacemanufacturinganddesign.com/article/giving-voice-to-augmented-reality/ Retrieved 24 October, 2019.
7. ActiveWizards (2018) "Comparison of the Top Speech Processing APIs." KDnuggets. https://www.kdnuggets.com/2018/12/activewizards-comparison-speech-processing-apis.html. Retrieved 24 October, 2019.