The 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
24-26 September 2015, Warsaw, Poland

# CleverNAO: The Intelligent Conversational Humanoid Robot

Jessel Serrano, Fernando Gonzalez, Janusz Zalewski
Dept. of Software Engineering, Florida Gulf Coast University
Ft. Myers, FL 33965, USA
fgonzalez@fgcu.edu, zalewski@fgcu.edu

***Abstract*—The objective of this work was the creation of a robotic system that any person could talk to in the English language, in particular, pairing an artificial intelligence algorithm that processes natural language with a physical robot that could synthesize speech. The result, called CleverNAO, is a successful combination of a chatbot application named Cleverbot with the NAO robot doing the speech synthesis. An attempt to include speech recognition was also made, with mixed success.**

***Keywords—robotics; humanoid robot; intelligent robots; robotic conversation; artificial intelligence***

## I. INTRODUCTION

"Can machines think?" This is the question asked by Alan Turing which has since spawned numerous, passionate debates on the subject of artificial intelligence [1]. It has also spawned the famous Turing Test, a test which determines if a particular machine (or algorithm) can pass as a human. Since its inception, the Turing Test has, in fact, been passed by a few artificial intelligence algorithms. Some of these algorithms represent the latest technology in terms of artificial intelligence.

Artificial intelligence (AI) is part of a broad field called cognitive science, which is simply a study of the mind and the way it works. For the purposes of cognitive science, artificial intelligence is defined as "a codification of knowledge will finally explain intelligence" [2]. However, when it comes to software engineering, the purpose of AI is to use knowledge to solve real-world problems. One of these problems, similar to the problem of the Turing Test, is how to make an artificial device or creature appear more human. To address this problem, a technology has been created called chatbots. These are AI algorithms that process natural language and, using the analysis that results from the processing, output an intelligent response. It is the utilization of these chatbots that is the main concern of this work.

In this project, a chatbot is linked to a robot so that the robot could verbally speak what the chatbot's response is to a human. The chatbot used in this project is Cleverbot [3], a chatbot created by Existor [4] and the robot is the NAO robot [5]. The intended result is an application, named CleverNAO, that facilitates a verbal conversation between a human and the NAO robot, which emulates Cleverbot.

The importance of this project can be paralleled to the historical and cultural importance of the Turing Test itself. Although the human participant is not going to be fooled by CleverNAO (due to the physical appearance of the NAO robot), such a conversation between a human and a robot signals a new era in robotics and AI.

The rest of this paper is structured as follows. Section II defines the problem and outlines the tools used for it. Section III presents the software solution. Section IV outlines the experiments, and Section V provides some conclusions and perspectives on future work.

## II. DEFINITION OF THE PROBLEM

### A. General Overview

When it comes to AI and natural language processing, one can distinguish six main steps required in having a computer participate in a conversation and to take action (Fig. 1):

- Speech Recognition: Acoustic speech signal is analyzed to determine sequence of spoken words.
- Syntactic Analysis: Using knowledge of language's grammar, sentence structure is formulated.
- Semantic Analysis: Partial representation is derived from the sentence structure to articulate the meaning.
- Pragmatic Analysis: Produces a complete meaning for the sentence that comes with contextual information, such as time, place, speaker, etc.
- Pragmatic Response: Using the complete sentence and its meaning, respond with a sentence that follows the line of thinking.
- Physical Emulation: Have a physical figure (robot) verbally speak the response.

As a result, the computer should be able to either respond in a conversation, perform an appropriate action, or make a decision. For example, this response can either be the output of a sentence or the activation of a switch. In the case of this project, the first four steps are utilized to produce an artificial intelligent entity that can give a

response to a person. However, in order to emulate a conversation between entities, two last steps are needed.

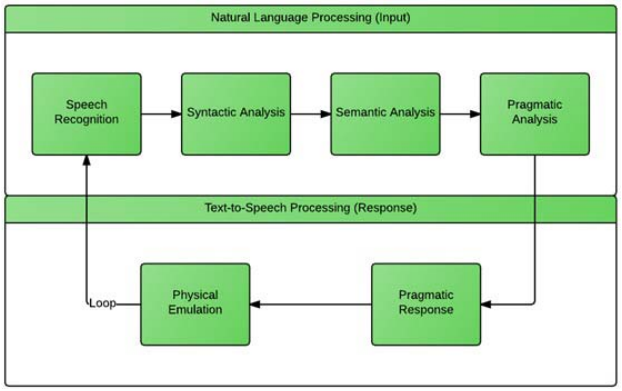


Figure 1. CleverNAO Conversation Cycle.

The first step is usually accomplished via an API that processes human speech via a microphone and translates the acoustic sound to sentences in a computer. These sentences are then passed to the second API which is the chatbot. The chatbot performs steps 2-4 through a series of HTTP connections and selected AI algorithms. Certainly, some of these steps, such as semantic analysis, are extremely difficult and have been studied for decades, but the focus of the current study is on using practical tools, rather than on theory, which is better explained elsewhere [6]-[7]. The final two steps are accomplished by having a robot verbally say the response. These six steps usually loop in order to produce a physical conversation between human and robot, as in Fig. 1.

### *B. Chatbot Tools*

A chatbot is a program designed to generate text that is imitative of human conversation. When using a chatbot, the user inputs the text via a keyboard, which eliminates the need for the program to recognize the acoustic speech signal. Instead, a syntactic analysis is required for the inputted text in which the program must begin to understand the sentence structure and grammar.

The chatbot algorithm being used in this study exhibits steps 2-4 continuously. It takes a String as a parameter and outputs a response that appropriately follows the conversational tone of the input String. This input String is the result of the first API which converts speech to a syntactic structure. The response is a String generated through an HTTP connection with a particular chatbot algorithm. For the purposes of this study, the chatbot algorithm in use is the Cleverbot [3].

Cleverbot is an artificial intelligence algorithm developed by the company Existor. Their goal is "to further develop our AI algorithms to achieve truly semantic understanding of natural language" [4]. Cleverbot learns from people and uses a database of millions of lines of conversations in order to produce a response that is likely to what a human would say. This algorithm allows it to grow and learn from conversing with humans much like an intelligent entity.

When selecting a response, Cleverbot searches its database three times in the online version. In order to pass the Turing Test, Cleverbot used 42 databases to produce a response that is semantically and syntactically correct as well as relevant to the conversation as a whole. It has been reported that Cleverbot passed the Turing Test by fooling 59 percent of people that it had conversations with into believing it was an actual person.

Once Cleverbot has been implemented with the Speech Recognition API referred to in the first step above (Fig, 1), the fulfillment of natural language processing for a program or entity is completed for each step. The result will be a response generated by Cleverbot. This response is then passed to a robot as a String, which is utilized via an API to convert text to speech. This robot is the NAO robot, described next.

### *C. NAO Robot*

The NAO robot is an autonomous, programmable humanoid robot developed by Aldebaran Robotics [5]. The NAO contains a plethora of sensors that can be used to detect motion and sound, as shown in Fig. 2. Its main connectivity is through either a WiFi module or an Ethernet port. This allows for software development over a network or via the Internet.

In this project, the NAO is manipulated via a network by a program on a remote computer. The NAO acts as the Server while the computer acts as the Client. This is seen by the way in which the NAO API is called; the API, on the side of the client, is called via calls to the respective methods on the side of the NAO's embedded API. The method on the NAO's side is what manipulates the NAO. All that is needed is the IP address and port number of the NAO robot.

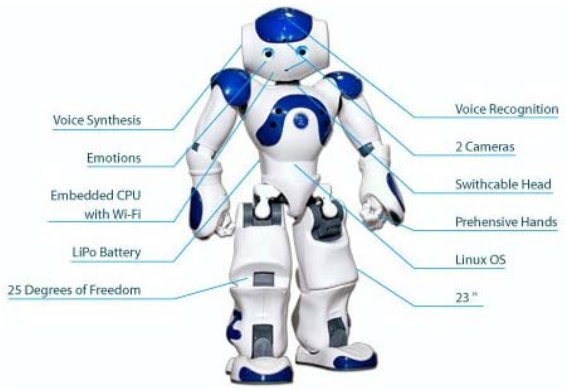


Figure 2. Sensors of the NAO Robot [5].

Using these two pieces of information, one can initialize a connection to the NAO robot by instantiating one of the modules of the NAO's library, NAOqi. In this case, the NAO's text-to-speech protocol will be manipulated so that the NAO could verbally speak the response of the Cleverbot. In this way, the NAO robot embodies the Cleverbot algorithm and becomes a part of the CleverNAO. The network connectivity is discussed in more detail in the next section.

## III. SOFTWARE DEVELOPMENT

### A. General Overview

The CleverNAO system is composed of six physical entities as shown in the physical diagram in Figure 3. Using a microphone, the user speaks a coherent and clear sentence. This speech is picked up by the microphone and sent to the laptop running CleverNAO. Using a speech API, these audio data are translated from speech to text. This process is essential to CleverNAO and is discussed in detail in the next subsection. Once the user's speech is converted to text form, the chatbot tool (Cleverbot) is to give a response back. This response is sent to the NAO robot, via the wireless router, in which the NAO will convert the text to speech and announce it via its speakers.

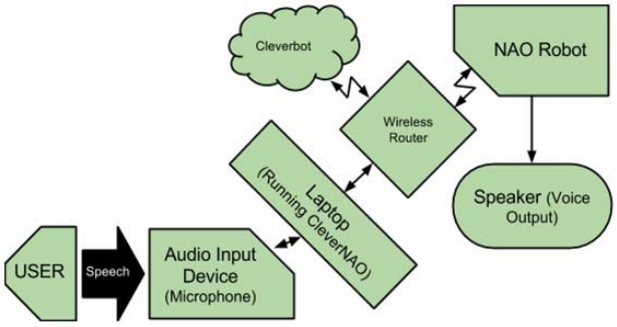


Figure 3. Physical Diagram of the CleverNAO.

As can be interpreted from above, CleverNAO is the main program that ties the conversation between the user and the robot. It controls three processes that are necessary for a conversation between the user and the robot to occur: speech processing, chatbot processing, and NAO processing. The name CleverNAO implies that the chatbot used in this project is the Cleverbot, as discussed previously. This process can be called "Cleverbot processing" to accurately reflect what is happening: using HTTP, a request is sent to the Cleverbot website from the CleverNAO program, to receive a response in return.

These three processes are highlighted in the context diagram, shown in Fig. 4. The CleverNAO acts as the main software that takes data from the user, the NAO robot and the Cleverbot website, and transforms the data to output responses which will be sent to the NAO and to the display or monitor. It begins with the user by verbal speech to the NAO. The CleverNAO will take in this speech and transform it into an input string using the speech API. The CleverNAO software will then send an HTTP request to Cleverbot.com. This request contains the input string and it asks Cleverbot to provide a response. Cleverbot.com will return with a response that is passed to CleverNAO. This response is also in the form of a string; this response string is then passed to the display for printing and to the NAO robot for processing. The NAO will take this response string and, using text-to-speech, speak the response to the user. This will loop continuously and the series of inputs and responses will lead to a conversation between the CleverNAO and the User, implementing a human-to-robot conversation.

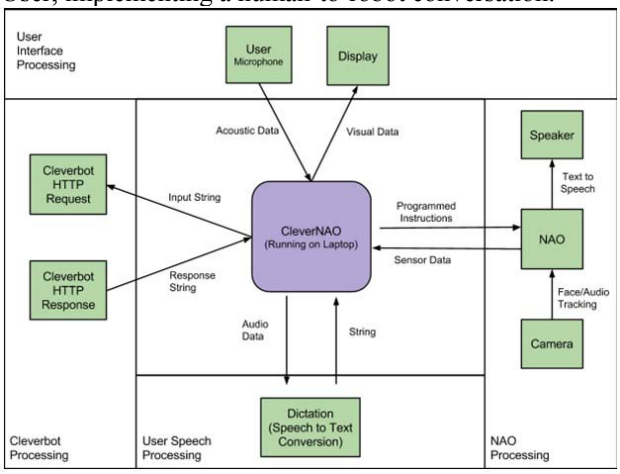


Figure 4. Context Diagram for the CleverNAO.

### B. Design Components

The CleverNAO program is created using Visual Studio and coded in C#. It is a Windows forms application that utilizes several APIs mentioned previously. The first is Microsoft's .NET speech library that has speech recognition (speech to text). The second library is an open source Chatter Bot API [8] that allows HTTP connections with three different chatbot algorithms: Cleverbot, JabberWacky, and Pandorabots. The final library is that of the NAO, which contains proxy classes that command the NAO robot. An overall flow of all three libraries and how CleverNAO uses them is shown in Fig. 5.

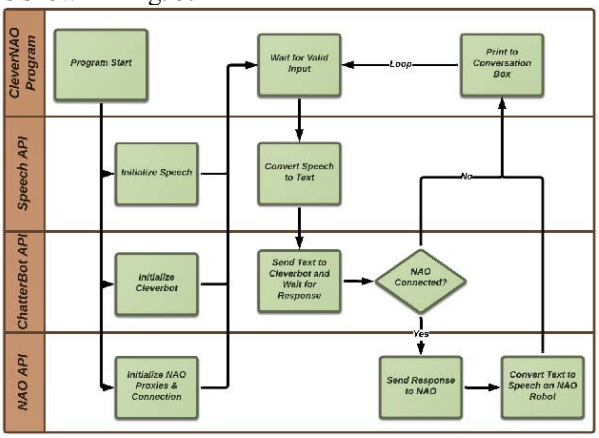


Figure 5. CleverNAO Program Flow.

*Microsoft Speech Platform*. The first process that occurs in the conversation between human and robot is to hear and understand what the human says. How this is accomplished in the development of CleverNAO is through speech recognition software, which takes in acoustic vocal sounds (verbal language) and converts it to text. Since development took place in Microsoft Visual

Studio, a native library is used to convert speech to text: the Microsoft Speech API, or SAPI [9].

Within the libraries of SAPI there is a Speech Recognition engine, as well as a Speech Synthesis engine. The latter converts text to speech and is not utilized in this project due to the NAO robot's text to speech capabilities. The Speech Recognition engine converts speech to text by using Grammar objects. "A speech recognition grammar is a set of rules or constraints that define what a speech recognition engine can recognize as meaningful input." [9]. Using DictationGrammar, a grammar that allows free text dictation, SAPI will try to recognize all audio input as speech and convert it to text. To ensure that the speech recognition does not continuously try to convert audio and background noise, a push to talk button is implemented.

Using the standard Windows Forms API, on a key/button press down, the speech recognition engine will begin to listen for speech to convert to text. Then, when the user has finished speaking, the key/button will be released and another event will trigger causing the speech recognition engine to stop listening and convert what it has heard to text, which will be outputted to the screen. This process is shown in the Fig. 6.

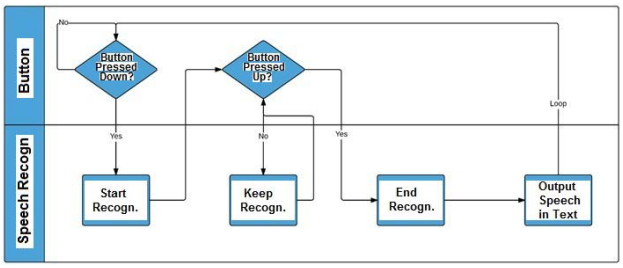


Figure 6. The Speech Recognition Process Flow.

*Chatter Bot API*. Another important process that occurs in the CleverNAO is the conversation with Cleverbot. As noted previously, this conversation is accomplished using an open source library [8]. This library has, within it, the ability to connect with Cleverbot through the Internet and receive a response directly from Cleverbot. It is here that the majority of the AI algorithm is executed, albeit indirectly. It is executed indirectly because Cleverbot is an already instantiated algorithm that is merely accessed through HTTP connections. What this library specifically accomplishes is the retrieval of a response that has already been built through natural language processing. A basic outline of this process is shown in Fig. 7.

As shown, several objects are created before an actual response is received. The Factory object first creates the chatbot object. This chatbot object is instantiated through a web connection with the respective chatbot algorithm, be it Cleverbot or another. This chatbot object then creates a new session which contains the main method used in receiving a response, Think(). The Think() method takes a string argument. This string is the input that comes from the user or the human. Through HTTP web requests and posts, the input string is sent to the respective chatbot algorithm, is analyzed via their server side, and returned to CleverNAO as a string. This string is the response to the user, to which the user must then choose to respond.

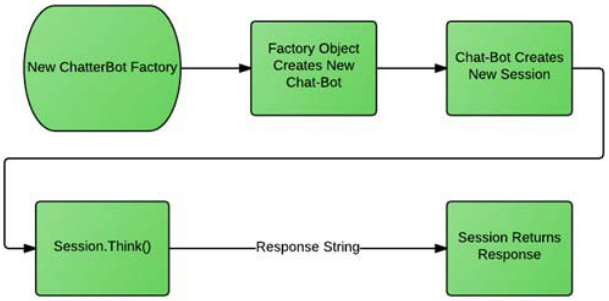


Figure 7. Chatter Bot API Process Flow.

*NAOqi Framework*. The final library used in the CleverNAO is the NAOqi library, which is the NAO robot's framework for controlling and commanding the robot itself. How the framework operates is shown in detail in Fig. 8 [10]. In this diagram, there is the Broker, which acts as the point of communication between the robot and the caller (the computer running the client). Located on the NAO robot is its own native library that contains methods which mirror what is contained in the Broker. This means that CleverNAO is not actually calling methods on the NAO, but rather calls proxy methods which it then passes to the Broker. The Broker then communicates to the NAO's native library and tells the library to call the methods to which the proxy methods correspond. This way, the CleverNAO can maintain a connection with NAO and, using the NAOqi Framework, command the robot with respective calls to its individual modules. These modules are separated by category. For example, there is a module for Motion, Audio, etc.

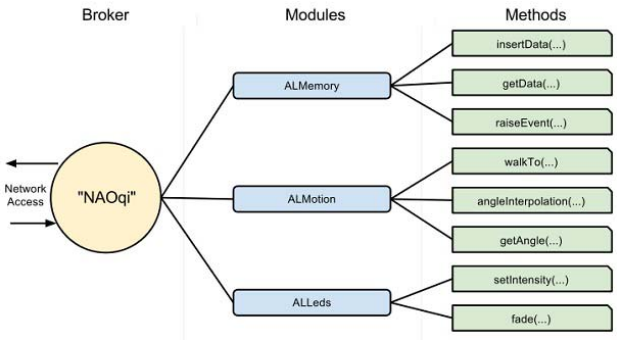


Figure 8. NAOqi Framework Overview [10].

The main module used in the CleverNAO is the Audio module which contains the text-to-speech proxy class. This class is instantiated using the IP address and

port number of the NAO robot. Once instantiated, its main method, say(), is called and is passed a string. This string is sent to the NAO via the Broker, in which the NAO will call its native text-to-speech library and actually speak the string that was passed to it.

## IV. EXPERIMENTS

This section describes documented experiments with the CleverNAO software and the conversations that result from a human-to-robot interaction. These experiments test the soundness of the CleverNAO system in two parts: without speech-to-text and with speech-to-text. The first experiment, without-speech-to-text, focuses mainly on verifying the soundness of the conversational capabilities of the robot and if a conversation can be carried out reasonably. This is to say that what is being tested is if Cleverbot can respond to user input without substantial delay, with intelligent responses, and with responses that follow the tone of the conversation. The second experiment, with speech-to-text, focuses on verifying the soundness of the speech-to-text and text-to-speech modules and whether, when implemented, they can handle a fluent and verbal conversation. This is to say that what is being tested is if user speech can be translated to text accurately and if the Cleverbot response can be translated to speech accurately.

Both experiments were conducted by running the CleverNAO program on a laptop that is connected to the wireless network that the NAO robot is on (Figure 9). In order to get a response from Cleverbot, sentences were typed into the input box. Upon pressing Enter, the sentences would be inserted into the Conversation Box and a reply from Cleverbot would ensue.


Figure 9. NAO Robot and CleverNAO Program.

The first experiment, without speech-to-text, was found successful. Using the input box on CleverNAO, text was sent to the Conversation Box in which Cleverbot replied directly (Figure 10). The responses by Cleverbot came within seconds. As shown in the timestamps in the excerpt of a conversation below, the longest response was a little more than ten seconds. As for the intelligence of the responses, they were on point and, at times, deeply philosophical. For example, when questioning the beliefs of Cleverbot, the response was, "I believe in other things, but not God."

The only issue with this experiment is the tone of the conversation. Occasionally, Cleverbot would randomly change topics and, when questioned about the previous topic, would reply as if it had no recollection of that topic that was just discussed. This issue is one of memory and reference and it may seem that this is a handicap to Cleverbot. To the CleverNAO system, however, this experiment demonstrates success.

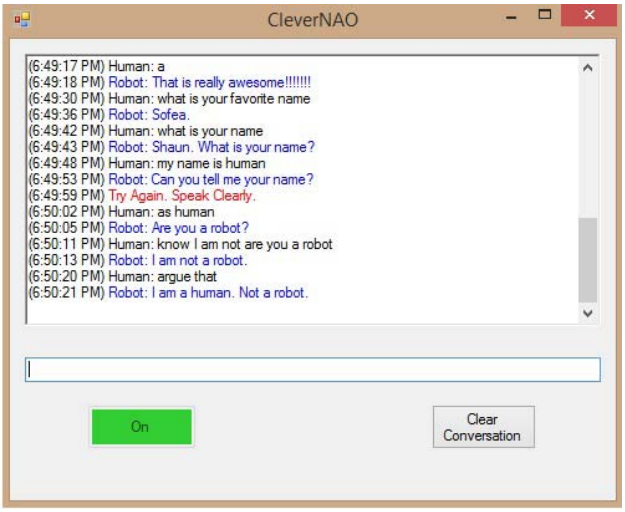


Figure 10. Push-to-Talk on CleverNAO Software.

The second experiment, with speech-to-text, was conducted similarly to the previous one. The main difference was that instead of typing a sentence into the input box, the sentence was spoken and then translated to text and finally sent to the Conversation Box to elicit a reply from Cleverbot. This was accomplished by holding down the Control key, speaking, and then releasing the Control key.

This experiment had mixed results. The latter half of the experiment, having NAO synthesis speech from text, was successful. The NAO robot enunciated words and phrases perfectly. However, the NAO robot failed when synthesizing expressions. For example, in the excerpt of one conversation, Cleverbot replies with a text phrase "*is a robot*", where the asterisks denote an expression implying an action or a result. In this case, questioning Cleverbot on its robotic nature resulted in Cleverbot revealing that it is a robot. The NAO, however, would synthesize this reply literally and enunciate the asterisk as "asterisk." The result is: "asterisk is a robot asterisk."

This minor issue, however, is overshadowed by the failure of the Speech Recognition engine used in this project, which fails in speech-to-text processing largely from the beginning. For example, when the first input was a spoken sentence, "Hello", it was translated to

"How old." From here, repeated attempts are made to speak "Hello" and have it recognized properly. Several inputs by a Human could be understood when analyzing it semantically, however, some were very off.

Consider this input: "aunt you said your name was for an." The actual spoken sentence was, "But you said your name was Frank." It would seem as if the speech recognition is finding difficulty in recognizing the beginnings and ends of a sentence most especially. Either way, this failure can be attributed to the CleverNAO system, yet only to its Speech Recognition module. In order to rectify this particular problem, either further development is needed to implement Microsoft's Speech Recognition or a new speech recognition library is needed altogether.

## V. Conclusion and Future Work

This project was centered on the creation of a physical robot that a person could talk to in the English language. The objective was to pair an artificial intelligence algorithm that processes natural language with a physical robot that could synthesize speech. The result, called CleverNAO, is a successful combination of a chatbot, Cleverbot, with the NAO robot doing the speech synthesis.

The CleverNAO application had to be built by using three unrelated libraries. The first is a Microsoft Speech library, used to capture speech and convert it to text. This allows the conversation between a human and a robot to be a spoken conversation as opposed to a written one. The second library is an open source library that creates an HTTP connection to several chatbot artificial intelligence algorithms. In this project, the chosen chatbot is Cleverbot. The final library is the NAOqi Framework and it is used to connect to and command the NAO robot. By combining these three separate libraries, an application has been created that facilitates and records a verbal conversation between a human and a robot.

The resulting CleverNAO application performs exactly as expected. Getting a reply from Cleverbot and having the NAO robot synthesize the reply to speech operates smoothly and concisely. The largest problem, however, lies with speech recognition. This problem is one of accuracy in the conversion from speech to text. Without a noiseless background, the accuracy of the conversion is not optimal at best. If any words are not enunciated clearly, the conversion produces different words or does not produce text at all. In order to improve the accuracy of the speech-to-text portion of this project, a different speech recognition library is recommended.

To have a physical robot that is not only able to move, but also hold a verbal conversation is akin to notions of science fiction and truly intelligent robots. One can imagine a future where these robots are the new store clerks, customer service agents, teachers, even friends.

Possible extensions of this project are to improve the speech-to-text process and to animate the NAO during conversation. The latter, animation of the NAO, can be implemented by using either the Face Tracking or Sound Localization modules in the NAOqi Framework. If implemented, these modules will allow the NAO to turn and face the person who is speaking to it, making the interaction more human.

## Acknowledgment

Part of this work has been supported by the University of Central Florida's NASA-Florida Space Grant Consortium (UCF-FSGC 66016015). Additional support has been provided by a grant from the National Science Foundation (Award No. DUE-1129437). Views expressed herein are not necessarily those of the funding agencies.

## References

[1] A. M. Turing, "Computing Machinery and Intelligence," *Mind*, vol. 49, 1950, pp. 433-460.

[2] E. Spyrou, D. Iakovidis, P. Mylonas, *Semantic Multimedia Analysis and Processing*, CRC Press, Boca Raton, Fla., 2014.

[3] *Cleverbot – Chat with a Bot About Anything and Everything*, URL: http://www.cleverbot.com.

[4] Existor, Exeter, UK. URL : www.existor.com/company-about-us.

[5] Who is NAO? Aldebaran Robotics, Paris, France. URL: www.aldebaran.com/en/humanoid-robot/nao-robot.

[6] J. Friedenberg, G. Silverman, *Cognitive Science: An Introduction to the Study of Mind*, Sage Publications, 2005.

[7] W. Minker, A. Waibel, J. Mariani, *Stochastically-based Semantic Analysis*, Kluwer Academic Publishers, Norwell, Mass., 1999.

[8] P. D. Bélander, *Chatter Bot API: A Mono/.NET, Java, Python and PHP chatter bot for Cleverbot, JabberWacky and Pandorabots*. URL: https://github.com/pierredavidbelanger/chatter-bot-api.

[9] *Microsoft Speech API (SAPI) 5.3*, Microsoft, URL: https://msdn.microsoft.com/en-us/library/ms723627(v=vs.85).aspx

[10] *NAOqi Framework*, Aldebaran Robotics, Paris, France. URL: http://doc.aldebaran.com/1-14/_images/broker-modules-methods.png.