CS Capstone Final Document

January 22, 2019

# How to Build an Effective Robot Comedian

Prepared for

Oregon State University

Prepared by

Group CS13

Short Circuit Comedy Club

**Abstract**

The purpose of this document is to outline the research papers that this team will create to conclude during Spring Term 2018. The three members of the *Short Circut Comedy Club* have spent their time during winter term perfomring research under Dr. Heather Knight at Oregon State University. The focus of this project is to study the effect a robot comedian can have on a crowd of humans. Kevin Talik's research has been spent understanding what a Comedian can do to "Adapt" to a performance. Arthur Shing has been studying the voice of the robot, and the difference between "Robot and Human" character. One final aspect of Stand-Up Comedy that we studied is "Crowd Work". Anish Asrani has spent most of his time developing spontaneous Crowd-Interactions during the set.

## CONTENTS

# 1 Introduction to Project

This project was requested by Dr. Heather Knight, a Computer Science professor at Oregon State University. Dr. Knight purposed this project to be a pilot research project, to investigate how aspects of comedy can improve a robot's charisma in Human-Robot Interaction (HRI).

This project, if successful, can benefit and influence the fields of comedy and HRI by showing how certain features of a robot comedian can be conducive to creating an entertaining encounter between a human and a comedian or robot.

In stand-up comedy, comedians have to rely on their scripted jokes and their ability to improvise to have a successful performance. In many ways, a robot interaction can be compared to a stand-up set. Within the context of stand up comedy, a robot must be receptive to an audience and tell jokes to make an audience laugh.

Why is the field of robot comedy relevant and worth studying? With automation slowly replacing menial tasks in society, such as an ATM or an automated cashier, interactions with bots are going to be much more common. These machines are less engaging to interact with, but require reciprocity to obtain a goal. While these machines do improve ease of use and convenience, people are not as expressive towards the robot. Even if the machine is unsuccessful in completing its task, expressive robots are received in a more empathetic manner by the users [1]. The users perception of a robots expression is more significant than its true internal state when seeking to generate engaging interactions [2].

Dr. Heather Knight noted the importance of the audience recognizing the performer as a social being, rather than as a dull object. In this respect, artificial social intelligence can bring greater engagement to the realm of the theatre. A socially intelligent robot comedian can be crafted by utilizing non-verbal gestures and by conveying a sense of character through spontaneous interactions with the audience [2]. Additionally, a study by Katevas et al. [3] evaluated the influence of non-verbal aspects of joke delivery. Knight also noted how physical presence and embodiment in a robot creates a more expressive and engaging interaction [2]. In another study done by Sjbergh and Araki [4], having a robot tell a joke was found to generate more of an audience reaction than having the joke read by a human. However, this study evaluated joke performance by a robot, but not an entire stand-up set. To extend on these lines of research, we intend to focus on the effect of verbally and physically expressing robot character qualities during a stand-up performance.

## 1.1 Team Members

The members of our team are Kevin Talik, Anish Asrani, and Arthur Shing. Kevin acted as team leader, and focused on creating the adaptive portion of the robot comedian. Anish focused on creating the audience sensing and crowdwork portion of the robot comedian. Arthur focused on creating topical variations of robot and human versions of jokes, as well as animations for the robot comedian. Our client, Dr. Heather Knight, supervised and worked along with us in many parts of the project. She often provided guidance or help in areas of development that we had trouble with.

# 2 REQUIREMENTS DOCUMENT

## 2.1 Introduction

The field of human-robot interaction can learn a lot from stand-up comedy. A stand-up performance has a basis of scripted content, from which the comedian delivers jokes to engage the audience. Good Comedians can read the audience and sometimes adapt their delivery based on the mood of the room [5]. In social robotics, when a robot shares a space with a human, an interaction can influence the people's opinions of the robot. Additionally, evident character traits presented (through dialogue and non-verbal motion) by the machine can anthropomorphize itself, making it easier and more enjoyable to connect with for the human [2]. The purpose of this work is to explicitly evaluate what aspects of a robot comedian's performance are most salient to human audience.

## 2.2 Previous Research

Research on the improvement of HRI is indispensable for our project. In Heather Knight's *Eight Lessons*, gestures, liveliness, and joke timing are all aspects that can be incorporated into the robot [2]. Relatable and appropriate gestures significantly helps improve communication between the robot and the audience. If the actions are predictable, humans can relate to the robot. When watching someone perform an action, the human brain maps the actions onto itself and simulates the action in the best way possible. This is a physiological experience that should be replicated by the robot in order to enhance relatability. Simplicity is important as well. [2]

In addition, Knight observed that having the robot portrayed as a living character rather than just an object that is kept up on stage improved the overall experience for the observers. Having believable interactions can enhance the feeling of a living character. The goal of the audience tracking using sensors is to maximize enjoyment. The enjoyment levels were be read by the robot and used to modify upcoming jokes [2]. Pausing and letting the audience laugh is vital as well. Starting the next joke too early can break the rhythm and leave the audience baffled. Looking around and body poses should be used to fill the pause [2].

In a previous study of robot comedy [3], Katevas found that when a robot engaged the audience through eye contact, the audience was more receptive to the performance. Eye contact from the robot is important, as it is a non-verbal cue for direct interaction. The audience members can identify that the machine is making an attempt to engage with specific members of the audience. We will investigate this further by having the robot perform various non-verbal and verbal interactions using sensors. The idea is to have the audience be a part of the performance even if they are not the ones performing. This can be accomplished if the robot is socially intelligent.

Other studies by Katevas et al. [6] that involved evaluating the social dynamics of a live performance by a robot have used SHORE™ vision framework software to analyze and detect faces in the audience. SHORE™ allows for facial expression recognition, estimated age, gender, and eye or mouth openings [7], giving the study a heterogeneous audience model. These allowed for the robot to interact directly with specific audience members. However, usage of SHORE™ involves expenses and funds that are unavailable to us, so we will encounter behavioral limitations dealing with a homogeneous audience model.

A study by Guy Hoffman [8] noted the importance of anticipation in human-robot interaction (HRI). The timing and meshing of anticipatory action and perception are a useful framework for HRI. The greatest challenge when designing a robot that will perform on stage is to enable the robot to be both - expressive and responsive. Robot models in the past have ended up on either extreme; they are either real-time and do not allow for continuous expression, or they are very animated but do not allow well times reactive behavior.

Researchers have also proposed multiple design patterns to promote sociability in Human-Robot Interaction (HRI). Some of these include having an initial introduction, some sort of didactic communication, including personal interests or history, and recovering from mistakes [9]. These patterns in design are proposed to allow for more effective and meaningful social interactions. While there is yet to be much data or research on the validity of these claims, they may still prove to be useful in guiding the designs of our project.

Robots utilizing non-verbal communication and statically written audience engagement have been attempted in robot comedy. In particular, Knight has observed the importance of character and spontaneous interactions in creating effective comedy [2]. However, there is little research on the actual effectiveness of character and spontaneous interactions [6]. This project will aim to examine the effectiveness of character and spontaneous interactions in robot comedy.

### 2.3 Hypothesis

Our research will be guided by the following questions:

1) How can the robot make the audience feel like a part of the performance?
2) How can the robot convey and a coherent and well-developed character?
3) How can the robot adapt and influence to the audience?

We hypothesize that comedy scripts with greater degrees of (1) crowdwork, (2) character, and (3) adaptiveness, will create more effective comedy, and have a more positive response from the audience.

#### *2.3.1 Crowdwork*

Keeping the audience engaged is vital in creating an entertaining performance. Katevas et al. had some success leading to a better audience response when using gestures and acknowledging the audience's presence. A successful performance manages the dynamics of various aspects of interaction to the benefit of both the performer and the audience [3]. Interaction with the audience will make the comedian robot feel more authentic, like an entity, and less like an object. Knight's research found that the robot's connection with the audience is stronger if the robot is able to convey social intelligence. This helps the robot display that it is not just an inanimate object [2].

#### *2.3.2 Character*

Expressing character will be understood from the robot under the ”theory of the mind” [10]. Understanding the agents meta-representation of a behavior is helpful for the audience in relating to the robot's intent, desires and knowledge. A robot attempts this to understand and relate its desires and intent better to the audience [11].

The appearance of character in a robot may create more effective comedy. Research has shown that expressive behaviors in a robot may cause interacting humans to favor the robot [1]. One could argue that expressive behaviors are behavioral actions formed by an inner character. While robots currently are not capable of having intrinsic character qualities, we postulate that a robot which behaves as if it has character could be more effective at engaging an audience. The task of conveying a sense of character may also benefit from social behaviors developed in researching the effectiveness of crowdwork, as social interactions may increase the sense of agency [2] and thus aid the audience in grouping behaviors as acts of character.

### 2.3.3 Adaptiveness

When the robot tells a joke, it needs to make adaptive transitions that correlates with the response. For example, if a joke does well and is received with laughter, the robot needs to time the next joke so that it is delivering it when the audience is ready, and can hear. However, if a joke is not well received by the audience, the time a robot needs to wait for the next joke will be different than if there is laughter. This is important for the effectiveness of a performance, as the connotation of the next joke is determined by the result of the previous joke; an audience that is told a bad joke will be hesitant to enjoy a joke if the previous jokes were bad. A robot comedian needs to be able to stay in a joke if the audience likes it, or address/recover the bad joke before starting a new sequence. Timing, or anticipation for a new joke, when coordinated correctly, positively influences the fluidity of the task (the performance) [8].

## 2.4 Research Approach

This project will be carried out in three phases; A learning and exploration phase in Fall term, a prototyping and testing phase in the Winter, and our evaluation phase Spring term.

### 2.4.1 Learning/Exploration Phase

This phase of our development will focus on understanding Social Robotics and the technology of the robot. The three of us will become familiar with stand-up comedy and the dynamics of an audience-comedian interaction. The NAO robot behaviors are programmed in the software Choregraphe, which has an API for python. We will test primitive scripts of decision making and non-verbal behavior. This is to learn how the coding environment works and to familiarize ourselves with hardware limitations. Additionally, we will learn to work with the sensors on the robot, and how they function (microphone, camera, etc). To become familiar with the format of a stand-up performance, we intend to study jokes and comedy devices.

### 2.4.2 Prototyping/Testing Phase

In the prototyping and testing phase, we will develop early sets for the robot. These implementations need to reflect and support our research questions. Crowd-work will involve audience sensing, as well as jokes that incorporate a measurement of response from the audience. Character implementation will involve testing the differences in effectiveness of robot vs human joke delivery, and the effectiveness of robo-centric jokes. As a stretch goal, we also hope to prototype and test the effectiveness of adapting a set to the audience, using intelligent calibration of the sensors.

These prototypes will be in the form of 3-6 minute set scripts with several variants in Choregraphe. The variants will help us evaluate the research questions and can be be tested in front of a small sample of humans, or in the form of a video recording. For example, the robot may perform in front of a handful of friends, or recordings may be to show online live viewers. Non-mechanical feedback from our testing will influence the direction of our prototyping, meaning that the implementation of our research questions will adapt according to the audience response. By the time this phase is completed, we will have working sets of robot stand-up.

### 2.4.3 Evaluation Phase

While doing the research, we will perform 6 shows with audiences ranging from 10-30 people. Tests in this phase will be at a greater scale and with a more realistic environment. Each stand-up performance, or set, will contain bits, or subsections of content that will be categorized as crowd work, characterization dialogue, and jokes. We will be testing on a live human audience to learn the effectiveness of each bit in a set. Based of the effectiveness of each set, we will modify the set and behavior of the robot. By the end of this phase, we hope to have a working, effective robot comedian.

## 2.5 Methods

### *2.5.1 Tests*

Our research questions will be how we evaluate the effectiveness of a robot comedian. We will create performances with varying degrees of (1) crowdwork, (2) character, and (3) adaptiveness. We will vary the presence of each of the factors of the performance to determine which factors are most influential to the comedian. To test crowdwork, we may create one script may include no references to the audience (low degree of crowdwork), while another may include many instances of interactions with the audience (high degree of crowdwork). To test the effectiveness of adding character, we may create a script with random jokes (low degree of character), and one with a coherent character throughout the set (high degree of character).

Likewise, testing the effectiveness of adaptivity will include scripts with no adaptivity and scripts that adjust to audience response.These scripts will be run on the NAO robot, and performed in front of a small audience in the testing phase, and later a larger live audience in the evaluation phase.

The audio sensors on the NAO bot will be used to evaluate audience reception to a joke, which the robot can then adapt to. The microphone on the NAO does not perform well when receiving input from a large audience, as it is designed to handle smaller scale interactions. NAO interprets speech with the ALSpeechRecognitionProxy and ALTextToSpeechProxy [12].

### *2.5.2 Metrics*

To evaluate the effectiveness of our robot comedian, we will measure the response of the performance with the audio captured during the set, and surveys for the audience afterwards. Measuring the audio will help us understand a broad audience response, as well as data for adaptive functionality. Surveys will give a more in depth information on factors of the show that are not covered by audio sensing, such as opinions of the perceived character and quality of the jokes. For example, to see if the robot's character was conveyed coherently, the audience will fill out a questionnaire prompting them to describe its character, as well as some humanizing questions, e.g. ”Would you invite this robot to dinner?” These responses will be used to study if the robot matched the expected persona and gauge how comfortable the people are with the robot.

As each set will derive content from the three research questions, we need to measure the effectiveness of each portion compared across the 6 performances. The survey that the audience members take after the show will gauge the response to subsections. Additionally, there will need to be questions to establish a pretense of how an audience member felt before the show, and how the performance has influenced their opinions of robot comedy. We want to see if someone who has seen a robot comedian would recommend the show to others.

The microphone on the NAO robot is designed for small environment settings. It will be difficult to distinguish speech from separate sources in a noisy environment [13]. Audio levels will be important data to collect from a crowd, where a louder crowd response could correlate to a level of enjoyment. However, a problem could be that a crowd could be booing very loud, and if we do not distinguish between different sounds that the audience can make, we could accidentally associate a negative response with a good response. The background noise could be very different depending on the room size. The density of people in a room making noise may return different audio level [14]. It will be important to test the effectiveness of a microphone to receive input.

### 2.6 Conclusion

A stage presence for a comedian is important because it connects the audience to the content, making it more effective than soulless delivery. A robot has a disadvantage in this; being soulless is the essence of being a robot. For a robot to establish a stage presence, the machine needs to make efforts to connect with the crowd, present a cohesive character, and dynamically adapt to a response. We think that a round character personifies a relatable agent for an audience. If a robot can give insight to it's desires, behavior, and preferences during a performance, the robot will humanize itself, connecting itself to the crowd.

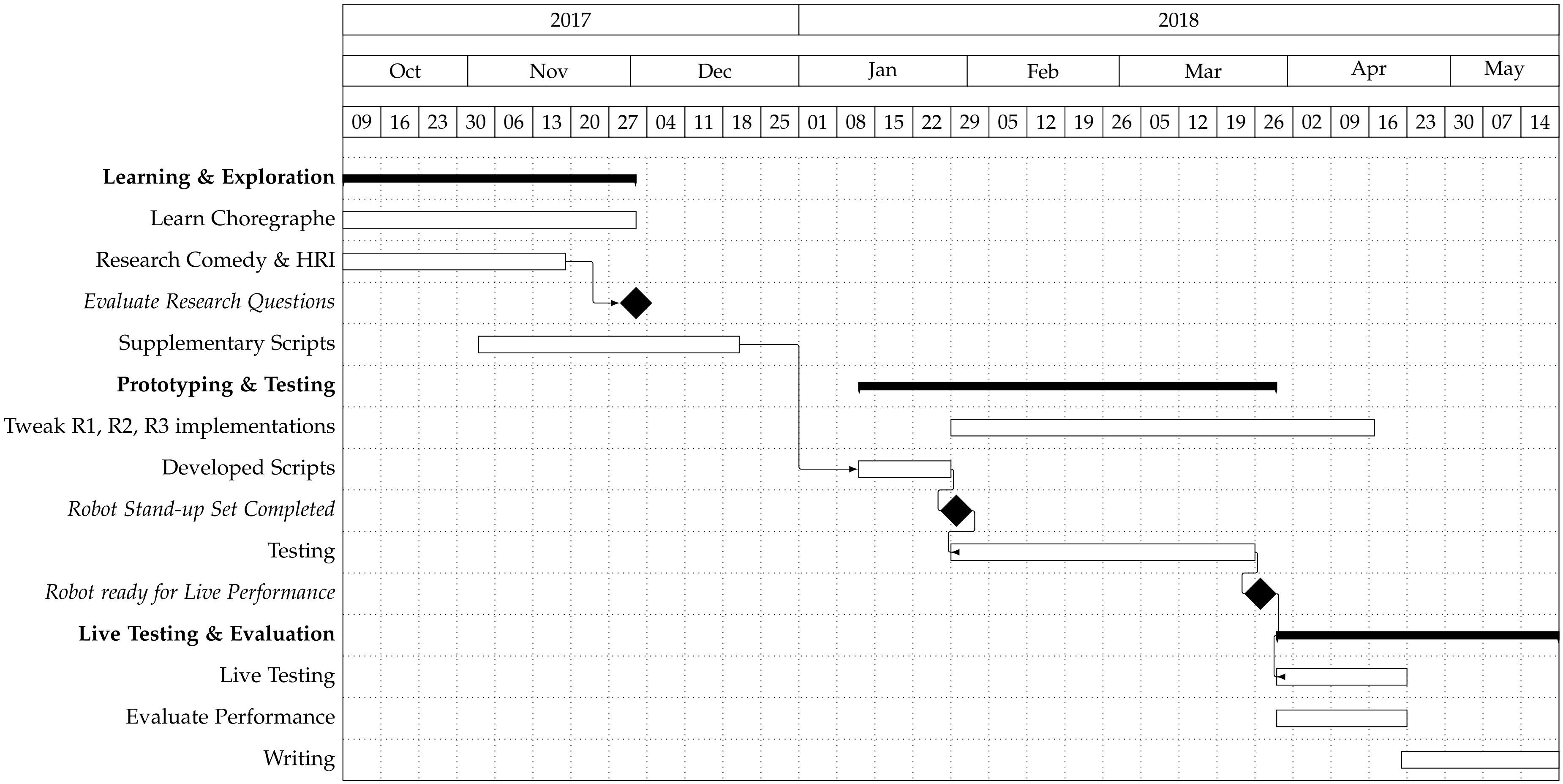



TABLE 1
A gantt chart showing the projected timeline of the project.

### 2.7 Modified Requirements

In terms of modified requirements, some of our existing requirements were deleted because they were infeasible to complete in time. Requirements that were changed include the coherent vs. incoherent character research area. This was modified slowly into the robot vs. human research area, as it was too hard to create an entertaining comedy set with an incoherent comedian. In addition, it became clear at some point in the project that doing multiple official tests would be infeasible, so we decided to do all our tests unofficially at the Engineering Expo.

## 3 Final Gantt Chart

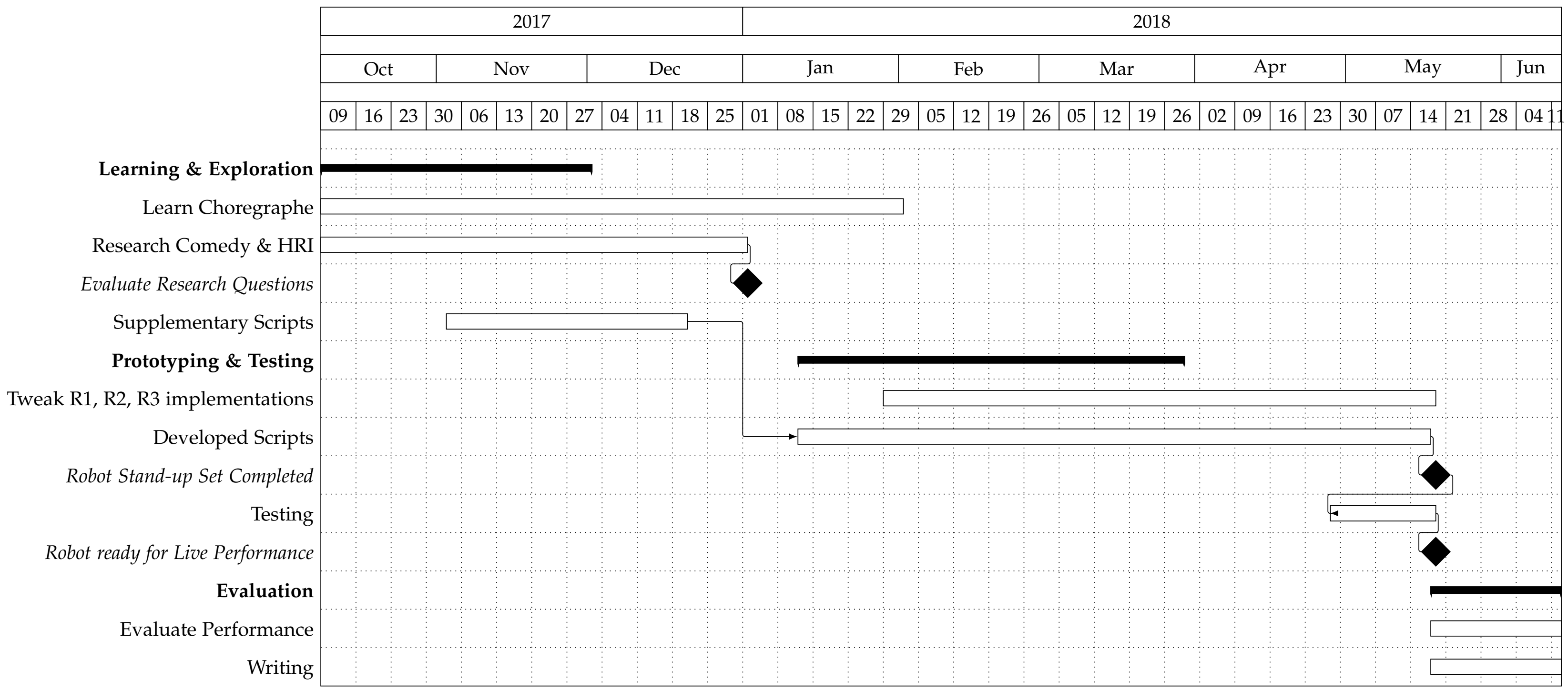



Fig. 1. A gantt chart showing the timeline of the project.

# 4 DESIGN DOCUMENT

## 4.1 Introduction

Comedy can come from robots of any kind. A carpet cleaning bot could miscalculate the end of a floor and tumble down a flight of stairs, or a voice-assisted tool might accidentally confuse a *Dinnertime Jazz* playlist for *Death Metal Essentials*. These actions that happen around a human can become shared experiences and references for the observer. For every shared task between a robot and a human, the human will often feel more empathetic towards a robot that makes an effort to be more aware of surroundings[1]. If the robot wants to become an effective comedian, it will have to attempt to be empathetic as well; it has to listen to the response of a completed action, and consider the change the bot brought to the setting.

### *4.1.1 Scope*

This paper covers the research questions that will guide the investigation of the interaction between a crowd and a comedian robot, as well as the main design goals for implementing comedy behaviors on a NAO Robot. There are three areas of intended research. These areas include the development of the intelligent adaptions to the crowd (adaptation), the integration of an audience into the performance of a set (crowdwork), and the exploration of robotic versus human-like storytelling, movement, and reaction to the audience (character). All of these components will culminate into the robot comedian, Ginger. At the end of the academic year, we will have a Robot Comedian that we have evaluated the effectiveness of. The main end-product is software for the variable stand-up comedy sets performed by the NAO robot, and the analysis of audience responses to our manipulations in audience adaptation, crowdwork, and robot or human-like character. Both of these will be described in our final paper.

### *4.1.2 Purpose*

The purpose of this document is to describe the development of the robot comedian, which involves: (1) the three main research questions, (2) the robot behavior implementations, and (3) the experiments we plan to use to answer the research questions.

### *4.1.3 Intended Audience*

This document is intended for stakeholders and developers in the research project *How to Make an Effective Robot Comedian*.

## 4.2 Research Design Overview

This subsection will cover the research questions for evaluating the effectiveness of a robot comedian. These questions will be the basis of the implementation of the comedian system. All three research questions will be presented including software requirements needed to answer the questions, and the experimental methods that will generate quantitative data about what impacts audience experience, participation, and reactions. The comedian system that is implemented will test three critical areas of a comedic performance corresponding to our research. The first question is about **adaptation** of a performance  how the robot and interpret an audience response. The second question studies **crowd work** during a show, and the choices a Comedian can make to engage the audience. Lastly, the third considers the implications of a perceivable **character** that the robot can portray, in particular robotic versus human-like.

### 4.2.1 *Adaptation*

4.2.1.1 Goal: We hypothesize that audiences will prefer a robot that acknowledges them, and integrates their data and responses into its set. To test this hypothesis we propose two tests: (1) to transition to topics dependent on the audience response, and (2) to present a crowd report upon completion of the set. The goal of this portion of the project is to determine if incorporating the audience into the set will enhance the overall performance of the comedian. To evaluate these hypotheses, we will conduct live studies in which people experience different versions of the software described below.

4.2.1.2 Methods: Generally, the performance will have two to three parts: the Seed Jokes, the Middle Content, and the Close. The Seed will influence the Middle Content (which will be chosen themed jokes and basis of the show). The Middle Content will transition to a intelligent or generalized Closing Joke when it is time to end the show.

Figure 2 depicts how a joke will be represented by the robot. It will perform the joke, collect audience feedback information, and branch to the joke that will best fit the response. At the end of the set, the robot will present a summary of what it thought that audience liked.

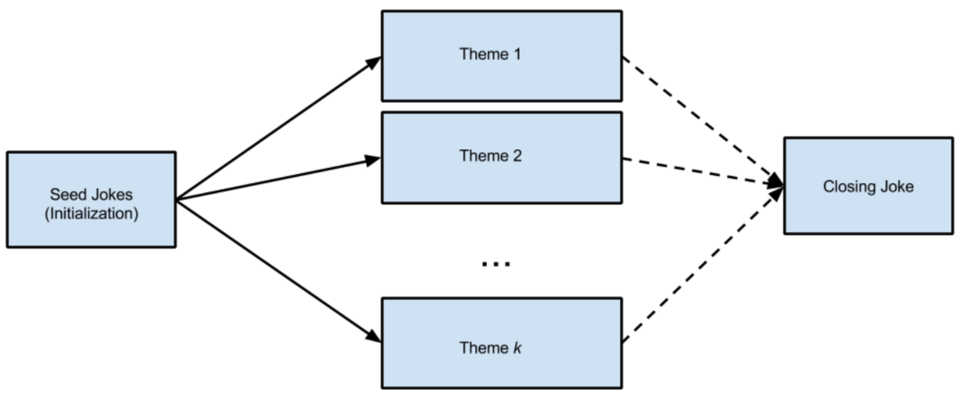



Fig. 2. This shows how the algorithm will have up to *k* Themes to choose from, determined from the seed joke. The closing joke is a subset of the set of all jokes, and may be outside of a specific theme. There could be different spanning trees of jokes that end at the same closing joke

In the beginning of the set, the comedian will present an initialization procedure, known as the Seed Jokes to test the response of the audience to different jokes. Depending on their response, the comedian will transition to a theme that is evaluated to be the best fit. The robot comedian will have many jokes to choose from that contain different material, but not all audiences will like all of the jokes. Figure 3 shows how the theme will be chosen from a set of up to k themes. From the Seed Jokes, one of the themes will be chosen. If there is time, we may also explore the choice of strategic closing jokes. These jokes might be stronger jokes than some of the others, and is helpful in ending the show on a stronger note.

For example, if two of the seed jokes are about food and Mindfulness, the performance will branch to the respective theme that matches the audience response (Branch 1 Food or Branch 2 Mindfulness). If jokes with a theme of food are not landing with the audience, the algorithm will need to know when to transition to a new theme, or when to end the set. When the robot tells a joke, it needs to be able to analyze the feedback and choose the next joke to perform. This needs to be done quickly, so that the robot is not spending noticeable time (for the audience) choosing a joke. There may not be a lot of jokes to choose from, but the choice needs to be made fast.

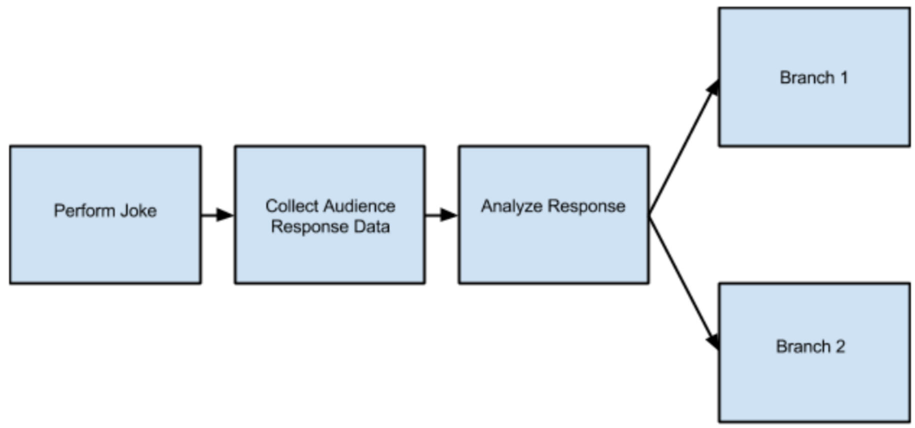



Fig. 3. This flow-chart depicts how, once the robot delivers a joke, will wait for feedback, interpret data, and then make a joke decision (Branch 1 and Branch 2)

The close of the robots act will include the robots report of what attributes the audience responded to most. The hypothesis here is that getting insight into the robots algorithms will increase the audiences perception of the robots intelligence, and, second, that it will make them laugh. People enjoy hearing about themselves.

All of the above behaviors, from adaptation to the end of performance audience report need to be evaluated with real people. Initial tests will be done on campus with small groups of people, the final test will be done in conjunction with the crowd-work and character manipulations, testing the entire algorithm together with a larger crowd, e.g., 10-25 people.

#### *4.2.2 Crowd-work*

4.2.2.1 Goals: Similar to the crowd report described above, we hypothesize that generally interacting with the audience (a.k.a crowd- work) throughout the performance, will improve the audiences overall enjoyment of the show. We want to analyze the

importance of this crowd-work relating to the central design of the project. Crowd-work should make the audience feel like they are a part of the show. This can be done in different ways - calling out and talking to the audience, watching the audience and incorporating them into the jokes, and asking them questions to keep them engaged, or to build off to make new jokes.

4.2.2.2 Methods: There are various kinds of crowd-work we want to test: one research question is whether crowdwork matters at all, and the second is does the crowdwork needs to be real or robot can just pretend it is paying attention?

To answer these questions we suggest three research conditions: (1) no crowdwork, (2) fake crowdwork, (3) real crowdwork. The first one would be no crowd-work whatsoever. The robot goes about performing its set and does not directly address the audience at all. The second could spanover-the-top and inaccurate crowd-work, or best-guess crowdwork, with the possibility of being real (e.g., predicting that most people in the audience were from Oregon, even if it did not hear what they said). The third case would integrate actual robot sensing. It would be important for conditions #2 and #3 to be parallel to assess whether crowdwork really matters.

As condition one is fairly obvious, let us discuss deeper possibilities for condition #2. In the obviously fake research condition, the robot will talk to the audience directly but it will be completely wrong in its observation. The absurdity of a robot trying to understand the audience and being completely off could be entertaining for the audience, or it may not connect with the audience at all. The exact reception of this sort of crowd-work is something we are trying to study. The other version of condition #2 is realistic but premeditated. For example, pre-known facts about the audience could be built into the robot or guessed. These pre-known facts could include the location of the performance, age demographics of the audience. For example, if the audience is known to be college-aged, the robot could be fed input to make comments about things relevant to college students.

Using actual robot sensing data is condition #3, and is certainly the ideal model, but requires sensing capabilities, processing power and hardware, so it would be good to know if it is really necessary. In this condition, the robot would be actually looking for cues from the audience during certain situations. For example, one example is asking questions and capturing words from the audience, then using that same word later. For example, the robot could ask a simple question about the weather, or the audience members hometown. In this case, the robot can listen for specific words and ask another question about that specific town or city.

Another real sensing capability the robot could use is audience volume levels after the delivery of jokes. The robot will keep track of the audience input. The robot could then acknowledge if the audience enjoyed the joke or did not enjoy the joke using these inputs. Additional sensors and processing abilities on the NAO robot include face-detection and bumper detection, so the exploration of audience sensing could potentially include speech, volume, vision, and touch.

All of these conditions will be assessed with live audiences (even if its just a few people in a classroom) to check to what degree is crowd-work important for a robot comedian. The audiences response will be used to see if they enjoy a humanized robot or if they prefer a more robotic one, or maybe even a combination of both. As crowd work is just a form of human interaction, we expect it will improve the audiences perception of the robots intelligence, add surprise to the show, and increase audience enjoyment levels. On the other hand, perhaps faking it can get 80the real version. That will be part of the evaluation.

#### *4.2.3 Character*

4.2.3.1 Goal: The goal of this subsection of the project is to examine whether or not robot comedy can benefit from having jokes delivered from a robots perspective. Our hypotheses are that a robot presenting jokes about technology or being a robot will be funnier than a human telling the same jokes, and that robots will be less funny than humans at telling jokes from a human perspective.

In Jerry Palmer’s *Taking Humor Seriously*, comic meaning is argued to depend on the interrelated factors of a joke’s context and setting, its delivery, the identity of the deliverer, and the audience [15]. Of specific interest to us are the factors of a joke’s delivery and the identity of the deliverer. In previous studies, robot comedy has been used to analyze effective aspects of joke delivery. However, little has been done in discovering effective aspects of a joke’s content as it relates to the identity of the deliverer. For example, Sjöbergh and Araki [4] found that jokes were perceived as funnier when delivered by a robot, rather than being delivered in text form. However, Sjöbergh and Araki used word-play jokes that were gathered from the internet, and delivered them through a robot by using a flat, machine-like sounding text-to-speech tool called AquesTalk. This form of delivery does not take into account the importance of effective joke delivery. While Sjöbergh and Araki did not implement measures for analyzing non-verbal delivery, other work has examined the importance of non-verbal signals in delivering jokes [6] [2]. Despite this, there is little to no existing literature on the effectiveness of jokes related to the identity of the deliverer. In our context, this means examining the effectiveness of robot-specific jokes in robot comedy.

4.2.3.2 Methods: To address this goal, jokes will be written from a human or robot perspective. The jokes written from a human perspective will have a corresponding robot version, ideally with as much one-to-one correspondence as possible in regards to cadence, length of joke, parallel content, similar motions, and so forth. These jokes will be subject to intense scrutiny by members of the project and by the client, such that revisions and edits can be made to create funny jokes with a definite correspondence between the two versions. For example, a human version of a joke might look like the following (lines with a definite correspondence with the robot version are highlighted):

```
Hey, hey, I got news. This is big.
Ok, quiet down. Get this.
That's RIGHT folks.
I'm no longer single. *throws hands up*
I met a man on tinder.
His name's Sebastian. He's a math nerd.
Swiped right as fast as my fingers could move.
```

Whereas, the robot version of the above joke is shown below.

```
Hey, hey, I got news. This is big.
Ok, quiet down. Get this.
That's RIGHT folks.
I'm no longer single. *throws hands up*
I met a robot on tinder.
His name's Data. He's a really geeky robot.
Swiped right as fast as my motors could turn.
```

4.2.3.3 Development process of joke writing: These jokes will be scripted in Choreographe, where adjustments to vocal tones and pausing will be made. Then, animating the robot for non-verbal gestures will be done to enhance the delivery. The overall process may look similar to Figure 4.

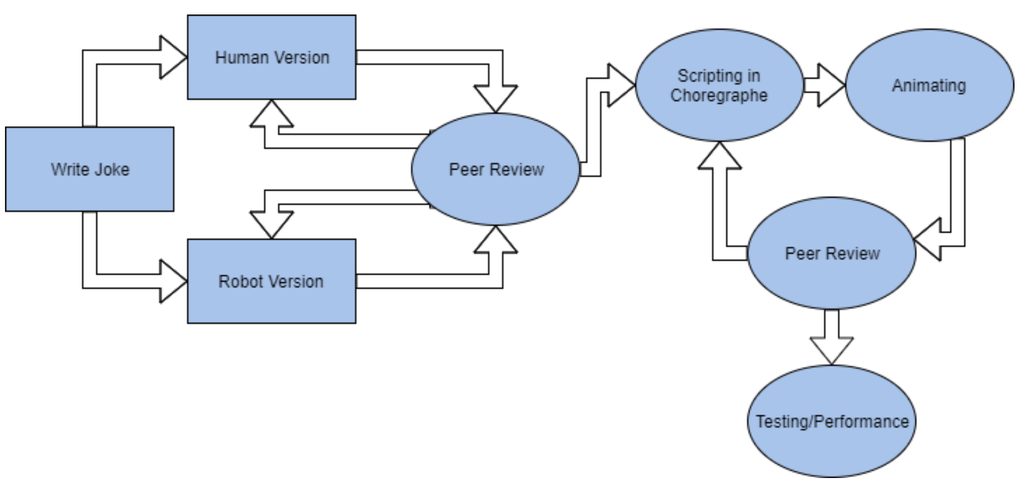



Fig. 4. The work flow from joke writing to testing.

4.2.3.4 Experimentation: The stand-up routine of the robot will comprise of text of the joke themselves, the motions that the robot uses to accompany them, and the way the robot surveys the audience after each punchline, e.g., in a human or robotic fashion. To determine the differences in audience response to the routines, studies will be done first on Amazon Mechanical Turk, and later with co-located audiences. Participants will be shown a video of the robots stand-up routine, and then presented with a short survey. The routines will be between 5 and 10 minutes long, and the survey will include questions pertaining to each joke or routine. Participants will be compensated with standard rates for watching brief videos and answering survey questions.

### 4.3 Conclusion

This document overviews our main Robot Comedy research questions, and the three software implementation targets for the capstone: adaptation, crowd work, and robotic versus human-like character. We hypothesize that all three will play a role in designing an effective robot comedian.

Designs for the adaptive transitioning algorithm between jokes, the integration of an audience into the performance of a set, and the exploration of robotic character in joke content and delivery have been described. To evaluate each, we have described the hypotheses we have and the research conditions we will compare to validate or invalidate them. People will be the ultimate judge of whether a robot performance will be successful, thus, human evaluations are critical. This project will use of combination of in-person and video studies to evaluate each research question individually in the winter term, then bring all parts of the programming together for collective evaluations with larger audiences on campus in the spring.

In following through with this in the research and development process, we will better understand the role of acknowledging and interacting with the audience, as well as the robot character itself, if creating experiences between people and robots, on and off the stage. Not all robots will tell jokes, but understanding more about what people value in robots could be reused in related robot applications from tour guides to English teachers, or even a factory robot that delivers parts and lightens a workers day by making jokes about her favorite sports team.

## 5 Tech Review - Kevin Talik

### 5.1 Introduction

To make an effective robot comedian, we have developed three research questions that will be the basis of three internal systems for the machine. First, we are investigating how spontaneous interactions benefit a set. Second, we want to quantify the benefits of the robot's ability to personify it's character, and lastly how to adapt it's set corresponding to the audiences reaction. My role in this project is to develop an algorithm to adapt the robot's set of jokes based off of audience response.

Timing and anticipation for jokes are crucial for the success of the set. Every joke that is told provides information to the audience about the robot, and from the opening joke; each new part of the set will build the repertoire of the bot. If a joke is well received by the audience, the relationship between the comedian and the crowd is strengthened, as the people will become more empathetic towards the robot. Every joke will enable the audience to connote decisions, preferences and knowledge of the robot. This is where the connection, otherwise known as the "Theory of Mind", is made with the audience[10].

This paper is an overview for the technology that will be used to create this algorithm. There will be a brief discussion on some of the previous work done in this field. Next, there will be a comparison of programming languages that will best suit this project, followed by a section describing some of the available tools for creating an Artificially Intelligent machine. Since audience feedback is a large portion of how this robot will attribute heuristics of the audience, there will be one section describing different ways to provide input from the robot.

### 5.2 Individual Role in Project

Our group will be working collectively to make an effective robot comedian, but I will be specifically working on our third research question: the robots ability to adapt a set based off of audience response. From audience response to a joke, or ”bit”, the algorithm should be able to determine the best fit for the next joke. This algorithm will continue to find the next best fit for a set until the 3-6 minute set is complete, or the robot decides to that the audience is done listening to jokes. The portion of the algorithm that tests the audience’s preference for humor will be known as the ”seed”. The seed will be a small subset of jokes, 2 - 3 jokes, that represent the collection of available jokes. For example, the seed bit may have three jokes; one joke could be a self-depreciating joke, one could be a joke about food, and another could be a quick observational joke about the audience. If the audience responded well to the self-depreciating joke, the algorithm should choose the next joke should be at the expense of the robot.

The seed of the set will give the audience pretense to the performance, and is the introduction of our robot. The delivery of the joke, and the content given has a large impact of robot character, and is out of the scope of the set creation; this is more suited toward the characterization research question. Also, the robot may need a more fluid way to interact with the audience (such as small talk, or crowd-work), which is underneath the breadth of the second research question about audience interaction. My contributions will be towards the system that determines which jokes, from our library of jokes, is best fit for our audience.

The algorithm will need to be able to input the strength of a delivered joke, and return a joke with attributes that match the strength of the joke. This will begin at the seed portion of the algorithm, and pick jokes until the set has lasted 3-6 minutes. The jokes will have some small variability in delivery, that correspond to the tasks of audience interaction and characterization.

### 5.3 Previous Work

There have been a couple of previous studies of robot theatre, most notably Dr. Heather Knight [2], Katevas et al [6], and Dr. Guy Hoffman [8]. To accomplish this task of designing an algorithm that can learn from the audiences’ response to jokes, it is important to look at previous research, as well as the tools available for accomplishing this task.

#### *5.3.1 ComedyParser*

Katevas et al [6] has researched a robotic comedian agent previously with some success. During their research, they implemented a program called ComedyParser (https://github.com/minoskt/ComedyParser ) that collects audience response information from SHORE computer vision, and performs the stand up set. The decision components, or what the robot does with information gathered from the SHORE vision, will be most important for implementing an algorithm in our project. A limitation with ComedyParser is that is specifically needs the SHORE vision to operate. SHORE will be too expensive for our project, and we will have to pursue more freely available systems. One solution that we had devised was to interpret the strength of the audience response to buttons on the robot. This will bypass the sensing component of comedy parser, as sensors for creating an audience model are out of the scope of this project. There is more discussion on on audience input in the ”Feedback Methods” sections below.

### 5.3.2 Anticipation in Robot Theatre

Research conducted by Dr. Guy Hoffman studying the implications of anticipatory actions in social robotics [8]. This particular study found that humans working with a robot that can monitor anticipation for an event allows humans to anthropomorphize the robot with more human like attributes. This study uses non-atomic Markov Decision Processes (MDPs) to model the decisions for events. Additionally, Hoffman models anticipation with an impulse-cue situation, where the robot is waiting for an impulse to trigger a specific cue. This is non-deterministic, as the MDP process is modeled around the probability of an event happening.

## 5.4 Programming Languages

There are three pieces of software provided by the manufacturer that are required to program the robot: Choregraphe, NaoQi, and Monitor. Choregraphe is the What-you-See-Is-What-You-Get (WYSIWYG) software used to visually program tasks as ”boxes”, that can be linked together to create sequentially or simultaneously executed behaviors. Each behavior is a series of tasks (boxes in Choregraphe) for the robot consisting of dialogue, movements, and sensing. NaoQi is the name of the developers SDK for programming tasks to the robot over text. It comes in many programming languages such as C/C++, Java, and Python. Choregraphe has boxes that allow only Python scripts to be executed [16]. Monitor is the monitoring software that monitors the memory allocation on the robot, as well as the camera [17].

This section will discuss the benefits and downsides to each programming language that are available for this project. There are seven SDK variations that support different languages; the most complete versions of the NaoQI SDK are, in order from most complete to least, are C/C++, Python, Java [16].

### 5.4.1 Python

Python is a translated language that is popular in scientific programming [18]. Pythons simple syntax and simple library integration make it quick to prototype and implement systems. Python is the second most complete SDK in terms of API functionality (C++ is the most complete) [19]. Python however is the only language that supports scripting from Choregraphe. Python is considerably slower than other compiled languages, and is not recommended for computations that are slower than 10 ms [16]. It is important to note that all of the C++ API functions are available from python as well [20]. Machine learning and A.I. are commonly implemented with Python, as such, there are a considerable amount of tools for implementing finite state machines and language processing. These tools are discussed below, in the ”Libraries and Artificial Intelligence Tools” section.

### 5.4.2 Java

Java is one of the least complete SDKs that is available for NaoQi [16]. Java is an object-orientated compiled language that is commonly used in mobile and app development. It is a higher level programming language than C++, but lower level than Python. Java works well with JavaScript, which can be used to develop HTML applications for controlling the robot (known as QiMessaging) [21]. This could be useful if there is a problem with using sensors to receive input from the audience. There could be more information fields in a web application that may represent the audience better than audience testing, and could leave us further away from the robot while it is performing. Java does not have Choregraphe support.

#### 5.4.3 C/C++

The C/C++ SDK of NaoQi is the most complete SDK [16]. C/C++ is an incredibly fast compiled programming language that has low-level programming qualities. Using C/C++ is required for loops that require faster than 10 ms response times [16]. The Aldebaran documentation specifies that this language is advised for advanced developers, and that Python be used before this. Additionally, this language does not have support in Choregraphe.

### 5.5 Libraries and Artificial Intelligence Tools

#### 5.5.1 Natural Language ToolKit (NLTK)

For Natural language Processing (NLP), the Natural Language Tool Kit is one of the largest, and most complete Python library [22]. Other libraries, such as TextBlob, build on and improve some of the functionality in NLTK to more specific uses (Textblob is elaborated below). In NLP systems, the largest components are grammar parsing and the collection of text (known as *corpora*). NLTK comes with over 50 corpora for building a domain dictionary, as well as an API for constructing context free grammars (CFG). Context free grammars are a set of rules that generate sentences. However, this model is dependent on a string input, which could also be used to model more abstract state machines that the performance set consists of. This is a large library, with much more non-specific tools to develop language processing. More specific tools, such as Pykov and TextBlob are more mature for certain tasks, and may be more beneficial to use in modeling state machines or generating large sets of text, respectively.

#### 5.5.2 PyKov: Markov Chains in Python

PyKov is a small Python module that is only for creating Markov Chains [23]. A Markov chain is a graph model with edges and nodes, where the edges are based on probability of traversing given edges. [24] is a simple visual explanation of how Markov Chains work. Markov Chains are different from CFGs in that a Markov Chain is dependent on the probability of traversing different trees through a graph; a CFG only accepts or rejects strings in a language, and may generate strings that are either incoherent or improbable. This is similar to how [8] models states in their robot that reacted to anticipatory events. Hoffman uses a Markov Decision Process, which is specific to an *agent* that interprets events and probability to traverse edges.

#### 5.5.3 spaCy

SpaCy is a large library for language processing, that is an optimized library using C modules [25]. The benefits of this library is that it has deep-learning integration, as well as part-of-speech string tokenizing. It would be a stretch to use deep-learning for generating sentences because it would take the focus away from the algorithms purpose, which is to choose the next joke from observations of the audience, and heuristics of a joke. Deep-learning is a feature we do not need to use, but this library has large functionality for input language processing. This library's speed is a benefit, as Python is normally slower than most compiled languages such as C/C++. If we need a response time faster than 10 ms, it might be more beneficial to use this library over TextBlobs, which is not optimized with C modules.

#### 5.5.4 TextBlob

TextBlob ”stands on the giant shoulders of NLTK” [26], but is specific to parsing ”textblobs”, and part-of-speech tagging. This library tokenizes text files, and creates dictionaries of seen words, as well as classifying words as nouns, verbs, adjectives, etc. NLTK can do this as well, but this module makes the task trivial. This could be useful in our project to specify word choice and building sentence data. However, it will be time consuming to manually find a corpus that conveys the same word choice and diction. Large corpora will be more difficult for our team to specify the importance of sentences and word choices. The other team mates are focusing on presenting a character from the robot, and smaller data could be easier for them to correlate their research goals to sentences.

## 5.6 Feedback Methods

### *5.6.1 Tactile Sensors on NAO Robot*

There are a few touch sensors on the robot, two on each feet, two on each hand, and one on the head. This would be the most simple solution to implement, as each button would correlate to an audience response. For example, if the audience likes a joke, I would touch the left foot, and otherwise I could touch the right foot. A difficulty with this approach is that there is a limited amount of buttons, and giving the robot a small input may limit the information the algorithm can use to make a decision. There could be variations of buttons to represent different input, but touching the robot all over during a set may be distracting for an audience.

### *5.6.2 Audio Sensors on NAO Robot*

Nao has 4 microphones on it's head, two on the front of the head, and two on the back. The sensitivity of each microphone is 20mV/Pa +/-3dB. This converts to -33.98 dBV/Pa. This means that a decibel noise that is 33.98 dBV/Pa when it is heard from the source will be picked up. The acceptable frequency range is 150Hz to 12kHz. This is a sensitive microphone, that would work better in small, controlled environments. The robot also can only pick up audio from one source at a time. Making this microphone work for a large audience would be difficult, and may be a challenge differentiating between a loud ”booing” sound, and a loud ”applause.” The benefit of using audio sensors is that we would not need to touch the robot while it is working, and could more naturally receive input.

### *5.6.3 Mobile Application for Operator Control and Audience Feedback*

The Javascript module QiMessaging [21] is used to send messages to the robot over Socket.IO [27]. This would be useful to remotely send signals to the robot by using an HTML5 webpage. Using this model, we would bypass the sensors on the robot and directly communicate from it. We would use our observations of the audience to relay to the robot for the algorithm to make choices on the joke. This would be useful to have, but would be time consuming to create. Depending on the time frame of the project, we may not have time to implement a system like this.

## 5.7 Final Thoughts

For this project, I think that using Python is going to be a valuable choice. Modeling automata such as Markov Chains and Context Free Grammars in Python is going to be more productive than other languages, Java being a second choice for NLP. Additionally, there are a litany of Natural Language Processing libraries that are written for Python that will be helpful if use use a large amount of grammar choices. Javascript will also play an important role to receive input about the audience, and since we will be the only ones viewing this webpage, the work that would need to go into making QiMessaging function would be not as significant if we made it public. All of the libraries needed to make CFGs and Markov Chains have benefits, and can be used at the same time. NLTK is very complete, but some features are not as fleshed out, such as PyKov in making only Markov Chains. While the English language is an important role in our project, using audio sensors may become a challenge. It would be the most fluid for a performance if the audience interacted fully with the robot, but covering audience interaction with the microphones on the NAO robot may be a stretch goal.

# 6 Tech Review - Anish Asrani

## 6.1 Introduction

During stand-up comedy, a performer's ability to influence the audience is vital to an entertaining performance. This involves correctly tying together various social signals such as body orientation, gesture, and gaze by both the performers and audience [3]. Dr. Knight emphasized on the importance of such non-verbal interactions in order to deliver a successful performance [2]. We want to add to this and incorporate crowd-work and audience interactions throughout the performances in order to help make the performance enjoyable and engaging for the audience.

## 6.2 Individual Role in the Project

One of the most important aspects of an effective performance is making the audience feel like they are part of the performance at all times. My role is to integrate this crowd-work throughout the performances. It can involve pointing at and talking to the audience which can be scripted, mostly during the intro and outro of the performance. It can be taken further by using sensors that will be triggered by certain audience actions and reactions.

## 6.3 Literature Review

The timing, frequency, and duration of a gesture relays a lot of information to the spectator. Dr. Knight conducted various performances with a robot, ranging from performing on stage with an audience to pre-mediated collisions with human environments, like "street performances." Over the course of these performances, there were eight major takeaways that will help make a robot performance entertaining. They were:

### *6.3.1 Convey Intentionality*

Using relatable and appropriate gestures help the communication between the robot and the audience. Since these actions can be predicted by the audience, they consider the robot as an entity similar to them. Displaying empathy while performing also boosts the robot's relatability to the audience.

### *6.3.2 No Mind Without Body*

Human expressions are derived from our physicality. Robots can also be capable of leveraging their embodiment to communicate on human terms. It was found that presence of a physical, embodied robot enabled more interaction as well as enjoyment of said interaction for humans. A robot not fully leveraging its physicality ends up losing a significant mode of communication and is also less expressive.

### *6.3.3 Physicality and Motion*

The audience should be able to connect the robot's non-verbal behaviors to the words. The human brain maps the actions it sees on to itself and imagines itself doing it. This helps the audience put themselves in the shoes of the performer.

### *6.3.4 Outward Emotional Communication Trumps Inward Experience*

The inner experience of the performer is trumped by the success of the outward intentionality conveyed. Most robots are designed to enhance, enable, or empower humans. Simplicity and clean physical design is often the clearest way to streamline communication of robot intention.

### 6.3.5 Gulf between Props and Character

Robots should be considered to be more like agents (entities) and less like props just standing up on stage. Until now, robots have been lacking in that department and there is a significant gap that exists. Robots lack believable and human-like actions. The various aspects of non-verbal communication like gestures have a significant impact on the robot not being considered an object.

### 6.3.6 Good Actors Outweigh Bad Actors

Multi-robot or human-robot teams have potential to deliver entertaining performances. Human actors can affect the audience's perception of the robot. They can also make up for the robot's unpredictability and lack of control.

### 6.3.7 Acknowledging/Learning

Human audiences are cognizant of human social behaviors. The audience can provide real time feedback. This feedback can be used to maximize the audience's enjoyment levels. The robot can constantly read these enjoyment levels and update the attributes of audience likes and dislikes. When delivering a joke, the audience should be given enough time to comprehend and process the joke. Starting the next joke early can break the flow and does not give the audience a chance to appreciate the joke and its delivery. The pause could be filled with the robot gazing around at the audience and posing. This helps develop a good rhythm for each joke.

### 6.3.8 Humor Makes People Like the Robot

Humor is one of the common grounds across all humans. When a robot performs comedy, and is able to match their sense of humor, it helps establish that common ground. If humor can help robot seem like ?one of us', that could be a significant leap to overcome the idea that robots are only props [2].

Katevas conducted studies in a similar fashion. His study hypothesized that interactional dynamics should be just as important to the mass interaction involved in performing comedy in front of a live audience. The interactional dynamics involved addressing the audience - including appropriately timed smiles or relevant gestures. The interactional procedures during a performance helps set the tone for the performance.

Robots provide a unique opportunity to experiment with the interactional processes. They can have a consistent routine while modifying the various aspects of delivery - body orientation, gaze, and gesture.

Embodied robots are likely the best way for the robot the catch the audience attention and make the audience pay attention to the robot. The robot used by Katevas et. al was a humanoid robot consisting of a robotic head, two arms with hands, the torso as well as two legs. The head had two rectangular LCD screens for eyes as well as LEDs on cheeks for expression [3].

## 6.4 Methods

Katevas' studies involved two performances. Each performance involved a compere doing the introductions and warming up the crowd for 10 minutes, followed by a human comedian performing for 13 minutes. Right after that, the robot comedian performed for 8 minutes. This format was used to widen the appeal of the event and to set up a stand-up comedy context. Each performance consisted of approximately 50 people in the audience. The sensors used to capture audience reactions and responses got data for approximately 20 people each performance. These sensors looked for various aspects about the audience including gender and age estimation. It also captured facial expressions and categorizing them into percentages of ”happy”, ”sad”, ”angry”, and ”surprised”. These facial expressions were used to update the audience model.

Punchlines were distinguished using a faster delivery followed by a short pause. The punchline was also followed by a gaze and a smile and sometimes laughter. The duration of the pause was determined by the feedback received from the joke - a longer pause if the audience is still laughing. While using gestures, a gesture pointing to the audience at certain times seemed to get the most positive feedback. The studies found that the enjoyment levels of the audience during the robot performance lied between the compere and the human comedian [3].

We could use the dynamic use of sensors to take into account and acknowledge when a delivery was successful or not. We will capture an initial audience model by performing a few ”trial” jokes to get a grip on what kind of humor or topics the audience would enjoy. The robot’s model for the audience could be acknowledged later to give the audience a feeling that they are being heard. While the sensors in our robot are not sophisticated enough to capture emotions from a crown of people, we will use audio feedback from the audience to update the audience model on the fly.

Dr. Knight’s research was more varied. Her performances ranged from stage and audience performances to guerrilla theater performances (street performances). The research looked to add value to developing everyday robots in addition to the entertainment value. It is easy to survey people and learn more about human-robot interaction when there is a significant amount of people present in the audience.

Using the theater context also helps the development of social robots. As noted before, non-verbal expression plays a key role in understanding sociability. A robot’s movement and engagement pattern impacts the people’s interpretation of the robot’s intention, capability, and state. Physical theater provides pre-processed methodologies for interpreting and communicating human non-verbal behaviors that can be portrayed on robots [2].

Capturing an audience model is not as feasible when performing guerrilla theater. The audience model would not be very accurate since the crowd is constantly changing. In such situations, the crowd-work would depend on gestures and being able to identify what jokes were successful.

### 6.5 Conclusion

Humor is one of the major emotions that is common ground for people from all backgrounds. Robots that have a grasp of humor would help bridge the gap present in human-robot interaction and bring people closer to robots. Stand-up performances provide a great platform for the robot to express its humor and show the audience what it is capable of. It will be vital to make the audience feel like a part of the performance, just like most successful stand-up comedians. In order to do this successfully, we need to take a lot from the research done in the past and build upon it.

## 7 Tech Review - Arthur Shing

### 7.1 Introduction

This is a tech review for Group 13, written by Arthur Shing, for the project ”How to Create an Effective Robot Comedian”. This project is a research project that intends to extend on research done in the field of Human-Robot Interaction. Our end goal for this project is to create an effective robot comedian that can perform in front of a live audience. In seeking to create an effective comedian, we hypothesize that comedy scripts with greater degrees of (1) crowdwork, (2) character, and (3) adaptiveness, will create more effective comedy, and have a more positive response from the audience.

#### 7.1.1 Role

As for my own role in this project, I am focusing on (2) the effectiveness of coherent character in comedy. This entails the creation of comedic scripts with varying degrees of character coherency.

#### 7.1.2 Technologies and Research Design

In the following subsections, I will explore the options and choices we had for alternative topics for our second research question on (2) character, robots, and SDKs and environments. Unlike the technology used for the other two research questions of (1) crowd work and (3) adaptiveness, implementing character will require fewer pieces of technology. For instance, both of the other variables will require some sort of sensing technology, while the implementation of character will not. Consequently, I had fewer technological features to choose from after research questions were decided. In place of the third piece of technology, I will discuss choices we considered for research questions.

### 7.2 Research Questions

The three research questions our group has decided on are the effectiveness of (1) crowd work, (2) character, and (3) adaptiveness in comedy. As a research project, these questions may be subject to change in the future. In deciding upon these three topics, various limitations were taken into account. In this subsection, I will discuss the decision on (2) character, as well as other topics we considered.

#### 7.2.1 Comedic Genre as a research topic

One topic we considered was the effectiveness of different genres of comedy. This includes genres such as deadpan, black comedy, slapstick, or specialized robot comedy. The first thing we considered was the feasibility of implementation. Deadpan comedy would either be the easiest or hardest part of the implementation, depending on the dynamics of the vocal software. The NAO's software is likely unable to mimic exaggerations in vocal expressions, but its monotonous (although mildly expressive) delivery may give way to a successful deadpan delivery. Black comedy is likewise feasible, although probably inappropriate for research. Slapstick humor would be harder to implement, with movement limitations that the NAO robot has. Similarly, the NAO robot would be limited by its battery life and its tendency to fall over. Specialized robot comedy is an area of great interest to us. This genre of comedy might include robot-specific jokes; jokes that only a robot would have the right to make. For instance, joking about its own programming, motor, or intellectual deficiencies might prove to be effective.

#### 7.2.2 Dialogical Comedy as a research topic

Another topic we considered was the effectiveness of two parties in stand-up comedy. In previous tests done by Heather Knight, it was found that a robot by itself could only maintain the attention of a human audience for 2-3 minutes [28]. However, with a human interacting with the robot, this time could be extended to twice or three times as long. Additionally, studies and research could be taken from ventriloquism. One example of a successful implementation of a robot comedian in tandem with a stand-up comedian in popular culture is Geoff Peterson from The Late Late Show with Craig Ferguson [29]. Peterson acted as a sidekick to Ferguson, and although he had an actual voice actor behind the scenes, Peterson could serve as an example of a goal for this topic. As for implementation on the NAO bot, most of the dialogue would be prescripted with lengths of time between each line for the human to speak. Another option would be for the human to hold a remote that activates the robot's next line, although this would seem much more scripted.

### 7.2.3 Character as a research topic

The effectiveness of character design was also a topic under consideration. In much of early previous work on robot comedy, jokes were simply read off one at a time, resulting in a rigid, static comedic structure [4]. Stand-up comedy, however, seems to thrive off of the buildup before each punchline, as well as the quirkiness of the comedian. Implementation of this may include writing scripts with varying degrees of character. Character might involve nonverbal gestures, expressivity, and consistency. For instance, a script with a high degree of character may mean more pronounced, expressive motions, as well as stronger expressive language. A script with a moderate degree of character may only include expressive language, or might even employ morose language to portray a different type of character. On the other hand, a script with a low degree of character may simply ramble off jokes, or speak with a minimal amount of non-verbal language.

### 7.2.4 Discussion of Research Topic Ideas

Overall, we found that the discussion of character as a research topic might prove to be the more fruitful endeavor. Additionally, aspects of comedic genre could be incorporated into the topic as a subtopic or stretch goal, seeing that certain genres of comedy necessitate certain comedians. For instance, robot-specific comedy may require a comedian with a consistently dry or self-deprecating character. Having the effectiveness of character as a research topic gives us more flexibility in adjusting our research questions in the future, as well as a broader range of ideas to draw from.

### 7.2.5 Conclusion of Research Topic Ideas

We will research the effectiveness of character implementation in robot comedy.

## 7.3 Robots

Our project will require a robot as the comedian to conduct testing. This robot ideally has vocal capabilities, as well as gesturing capabilities. The three following robots include both of these capabilities.

### 7.3.1 RoboThespian

The RoboThespian has been implemented in Human-Robot Interaction studies in the past. It is a humanoid, life-sized robot designed for human interaction in public. It was used to great success in stand-up comedy performance study done by Katevas et. al [6]. Additionally, it has been used in other theatrical contexts [30] to some success. The RoboThespian is constructed by Engineering Arts, and comes with a touchscreen kiosk for the user that can be used to trigger content, view sensors, and actively engage or control the robot. For the most part, the RoboThespian uses predefined timed animations in its performances through its stock software [6]. However, its responsiveness can be optimized in a Python IDE that is provided by Engineering Arts. The RoboThespian stands at a height of roughly 6, and weighs 97 lbs including a floor base. The robot also features pneumatic and DC servo motor actuators for the upper body, upper limbs, and head, including jaw actuation and distinct fingers. It also has a starting price of £59000 [31].

### 7.3.2 Pepper

The Pepper robot is a small humanoid robot that was built to perceive and respond to emotions. Designed by Aldebaran Robotics, the Pepper bot was created to interact as natural and intuitive as possible. It includes 4 directional microphones on its head, as well as two HD and one 3D camera to locate and identify emotions in voices and faces. It also comes with six laser sensors, three obstacle detectors, and two ultrasound sensors. The Pepper robot has also had experience in stage work and has been used in Japan to mild success [32]. It specializes in socializing and reacting to perceived emotions. The robot also comes with a tablet on its chest to supplement human interactivity. The Pepper weighs in at 62 lbs. It has a starting price of $25,000. However, when it was first released, it had a starting price of $2,000 with a $300 monthly maintenance fee. A Pepper SDK for Android was released just this year. [33]

#### 7.3.3 NAO robot

The NAO robot is a humanoid robot featuring upper and lower body joints. Also designed by Aldebaran Robotics, it has 25 degrees of freedom and multiple sensors for perceiving the environment. These include 4 directional microphones, as well as two HD cameras for vocal and facial recognition. The NAO robot was created to be extremely customizable. It has a small frame, weighing 9.5 lbs. It has been previously used to study Human-Robot Interaction and comedy by Knight et al. [34]. The robot can be programmed through a software produced by Aldebaran Robotics, called Choregraphe. It has a starting price of $9,500 [35]. [36]

#### 7.3.4 Discussion of Robots

The NAO Robot is the most accessible to us, as our client already owns one. The RoboThespian and Pepper robot have also only recently launched SDKs for development. This led us to be wary of possible bugs and issues that have not yet been addressed in development. Our client is also familiar with the development environment for the NAO robot, which will make learning it an easier process. In addition, the RoboThespian and Pepper robot are priced much higher and lack lower body mobility. The lower body mobility of the NAO robot may benefit from using non-verbal gestural cues that cannot be done on the other two robots. These gestures may increase the effectiveness of our comedy [2]. The RoboThespian and Pepper Robot are also significantly heavier than the tiny NAO robot, which may impact our efficiency with hands-on programming and transporting the robot.

#### 7.3.5 Conclusion of Robot Choice

We decided to use the NAO Robot for our project, based on accessibility and efficiency.

### 7.4 SDKs and Environments

The NAO Robot can be programmed in several different ways and comes with several different SDKs. It uses an API called NAOqi that is available in eight different languages. However, only C++ and Python are supported on the robot, while the others are only supported on the computer for remote access [37].

#### 7.4.1 C++

The C++ framework is the most complete one, and allows a developer to write real-time code at high speeds. The developer can access specialized proxies, which are optimized and give direct access to existing methods. These proxies also provide compile-time type checking. The following is an example of specialized proxy usage.

```
#include <alproxies/altexttospeechproxy.h>

const std::string phraseToSay = "Hello world";
AL::ALTextToSpeechProxy tts("nao.local" , 9559);
tts.say("Hello world");
```

In addition to these optimized proxies, generic proxies can also be used. With these, methods must be specified by the developer, which may be more prone to err. These are slower than specialized proxies but can adapt to any module. This includes user-created modules. User-created method names cannot clash with names bound by ALModule methods. The following is an example of generic proxy usage. [38]

```
#include <alcommon/alproxy.h>

const std::string phraseToSay = "Hello world";
AL::ALProxy proxy("ALTextToSpeech", "nao.local", 9559);
proxy.callVoid("say", phraseToSay);


// Or, if the method returns something, you
// must use a template parameter
bool ping = proxy.call<bool>("ping");
```

Movement programmed in C++ is done by calling the angleInterpolation method under the ALMotionProxy module. This is used by creating an ALValue::array of target angles and times, and passing them into the method. The following is an example of using angleInterpolation to move the NAO robot's head.

```
/**
* Copyright (c) 2011 Aldebaran Robotics. All Rights Reserved
*/
#include <iostream>
#include <alerror/alerror.h>
#include <alproxies/almotionproxy.h>

int main(int argc, char* argv[]) {
  const AL::ALValue jointName = "HeadYaw";
  AL::ALMotionProxy motion(argv[1], 9559);
  /** Set the target angle list, in radians. */
  AL::ALValue targetAngles = AL::ALValue::array(-1.5f, 1.5f, 0.0f);
  /** Set the corresponding time lists, in seconds. */
  AL::ALValue targetTimes = AL::ALValue::array(3.0f, 6.0f, 9.0f);
  /** Specify that the desired angles are absolute. */
  bool isAbsolute = true;
  /** Call the angle interpolation method. The joint will reach the
  * desired angles at the desired times.
  */
  motion.angleInterpolation(jointName, targetAngles, targetTimes, isAbsolute);
  exit(0);
}
```

### 7.4.2 Python

The Python framework is the second most complete, and allows a developer to run embedded code. Unlike C++, Python does not have two different kinds of proxies. However, methods can be directly accessed once an ALProxy is set to a module. The Python API allows a developer to use all of the C++ API from a remote machine. We can infer that this includes user-created modules. Programming movement in Python is also done by calling the angleInterpolation method under the ALMotion proxy. Unlike C++, Python does not require the usage of AL::ALValue::array, and uses its built-in lists. The following is an example of moving the NAO robot's knee in Python. [39]

```python
import sys
import math
from naoqi import ALProxy

def main(robotIP):
  motionProxy = ALProxy("ALMotion", robotIP, 9559)
  # KneePitch angleInterpolation
  # Without Whole Body balancer, foot will fall down
  names      = ["LKneePitch", "RKneePitch"]
  angleLists = [ [0.0, 40.0*math.pi/180.0], [0.0, 40.0*math.pi/180.0]]
  timeLists  = [ [5.0, 10.0], [5.0, 10.0]]
  isAbsolute = True
  motionProxy.angleInterpolation(names, angleLists, timeLists, isAbsolute)

if __name__ == "__main__":
  robotIp = "127.0.0.1"
  main(robotIp)
```

### 7.4.3 Choregraphe

Choregraphe is a multi-platform desktop application that is designed to create and test animations and behaviors. Unlike an SDK, Choregraphe allows for development with significantly less coding involved. All of the NAOqi API is accessible in Choregraphe, and animations can be done in an intuitive way. Each joint of the NAO robot can be virtually manipulated in a control console. Figure 5 shows the ways that the left arm of the robot might be manipulated in the console. These manipulations require no coding and allow for quick and intuitive viewing of movement options.

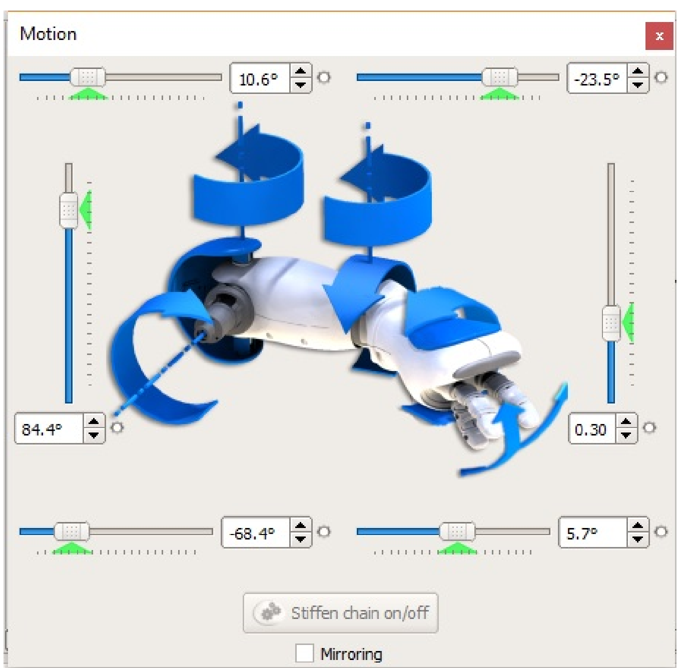



Fig. 5. The motion box in Choregraphe allowing for joint manipulation in the left arm.

Choregraphe can also connect to an actual or virtual NAO robot as shown in Figure 6, allowing programs to be run on the spot.

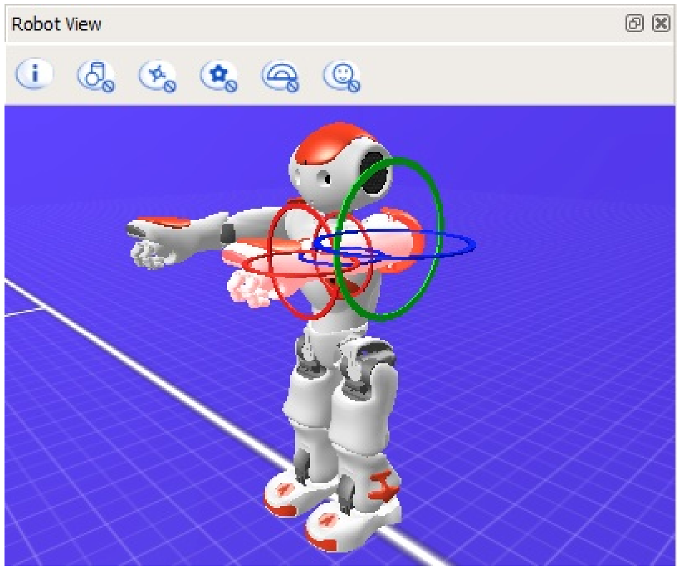



Fig. 6. The virtual robot in Choregraphe. In this image, the left arm can be manipulated.

Animations can also be created in tandem with vocalizations using a timeline feature, as shown in Figure 7. This feature allows a developer to choose the frames at which the NAO robot will position itself, as well as the words that will be spoken while doing so. [40]

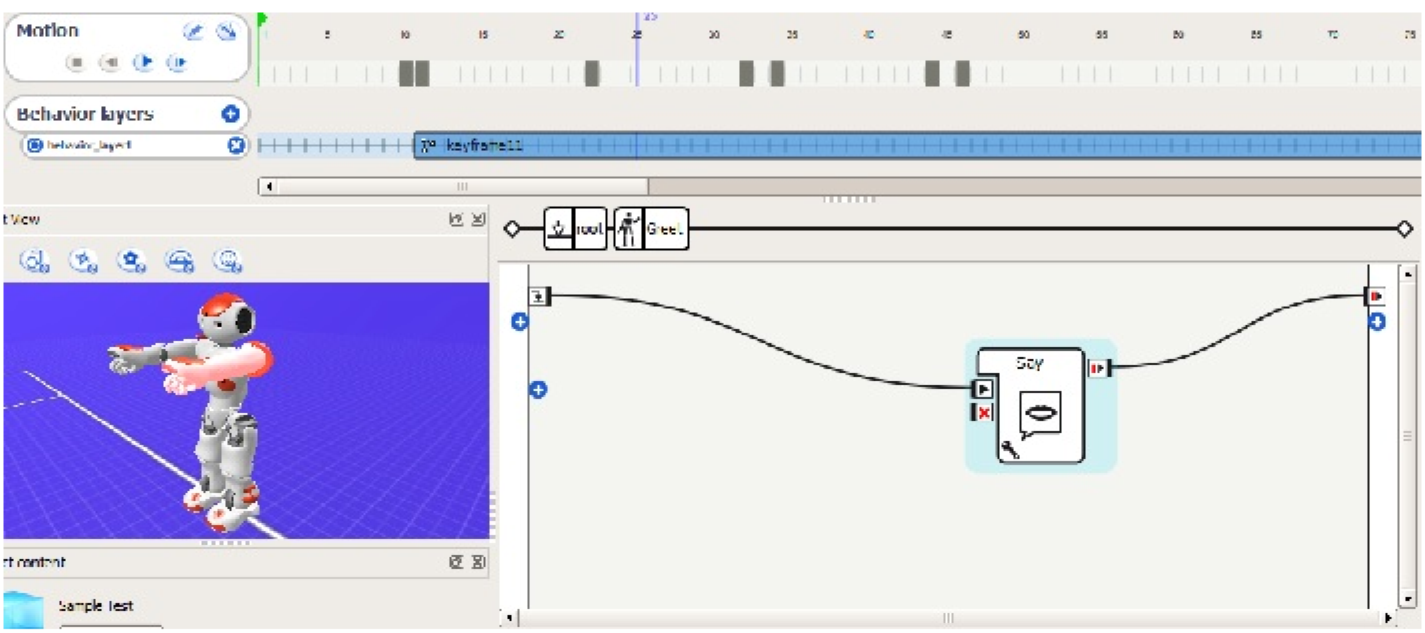



Fig. 7. The Choregraphe environment. In this image, the timeline is being manipulated for animations along with vocalizations.

#### 7.4.4 Discussion of SDKs and Environments

While the C++ and Python SDKs are comprehensive and well made, for my purposes, animations and dynamic actions are a greater focal point. Because of this, Choregraphe seems like the ideal environment for the research question of the effectiveness of character. As most tests will involve predetermined scripts and little sensor usage, movement will be a vital part of conveying certain character traits. This will be most efficiently done in Choregraphe.

#### 7.4.5 Conclusion of SDK and Environment Choice

For our purposes, Choregraphe will be used to program the robot.

# 8 WEEKLY BLOGS

## 8.1 Kevin Talik

### 8.1.1 Fall Term

- Week 1-2

  Got project this week; Problem statement due October 10; email client and team . We set up a slack channel ( akarobotics.slack.com ), and shared our availability.
- Week 3

  We need to come up with a proposed solution that is specific enough for us to define what we intend to fix, but vague enough so that we can have some room for exploration (as this is a research project). We need to make sure that our problem statement reflects that stand up comedy is a metaphor for human to computer interaction.
- Week 4

  Make sure github is organized and easy to understand for Kirsten and Ben. LaTeX Font, IEEE standard font for the first page. Sans-serif not serif
- Week 5

  - **Plans**

    Finish Scripts for local, remote and real world. Explain research questions in scripts Practice the nltk and choreograph
  - **Problems**

    Kirsten gave us a 92 on the problem statement, she is having difficulty understanding our research based goals.
  - **Progress**

    We are behind on the requirements, and need to focus on how this assignment will drive the research for the project. We need to focus on documentation quite a bit as well. Improve notes
  - **Summary**

    Meet with Kirsten to help elaborate on research deliverables. Organize research, look up ieee research.
- Week 6

  - **Plans**

    Kirrsten is looking at our rough draft right now, and there will also be a rubric for the final that we can look at. Research Class this week. We talked about:What are you trying to solve? Can a robot dynamically interact to a crowd? Robot Character perceivable to audience? Adaptive content based from audience feedback?

- **Problems**

  Expectations are a medium concern. Research based work is something we dont have to worry about without an IRB.

- **Progress**

  Ben mentioned that our project is difficult, and is masters and PhD level work. Mcgrath and Kirsten are meeting with Heather to make sure that we have a reasonable amount of that can be completed during this school year.

- **Summary**

  Our secondary objectives are going be structured around the dynamic nature of the crowd adaption. If we can come up with a clever solution to implement decision and dialogue choices, we can add it into our design of the machine, but we use the machine to perform the research

- Week 7
  - **Plans**

    Tech review, look at technology for language processing.
  - **Problems**

    ipsum
  - **Progress**

    Markov Decision Process, MDP

    https://www.cs.rice.edu/ vardi/dag01/givan1.pdf

    Pykov, markov chains in Python

    https://github.com/riccardoscalco/Pykov

    A way to make state machine with probability transitions, based off of obvservations
  - **Summary**

    Further Designed ways for crowd to receive input.ould test with a couple people yelling at a robot, to test what high audio levels do to the input

- Week 8
  - **Plans**

    Use tech review as both tech and literature review. Make sure that my choice of tools is unbiased. None of the ”i did it because i did it” sort of things.
  - **Problems**

    Current view of the tech does not describe the technology well with the features.
  - **Progress**

    Added more AI reviews of tech that I can use: Tensor Flow, pytorch, pykov, NLTK, CFG.
  - **Summary**

    Looked at feedback methods of audio sensors, and AI tools. Language processing may fall out of scope.

Its a lot of data. Use timelines to implement jokes, break jokes into one sentence in each box, put waits and ambient movement in each joke. Make two timeline boxes to vary the joke_robot and joke_human

- Week 9
  - **Plans**
    testt Say and branching dialogue in choregraphe
  - **Problems**
    Organizing large choregraphe functions (timelines are going to be better for things that depend on time, obviously you dunce)
  - **Progress**
    Got say boxes and branching to work, but in a messy way
  - **Summary**
    Movements and jokes are going to be best implemented in choregraphe. Algorithm may work by just taking text input, returning text input and the branches just follow the path
- Week 10
  - **Plans**
    Finish progress report
  - **Problems**
    We should be worried about the progress report, just because it is a lot of work due in a short time.
  - **Progress**
    Finished Progress report, wrote some comedic devices for comedy. Like, observational humor, Jobs jokes, like uber driver, dance lessons, unemployment
  - **Summary**
    Further looked at types of jokes, finished Progress report

### *8.1.2 Winter Term*

- Week 1
  - **Plans**
    Meet with heather, plan implementation term.
  - **Problems**
    Branching will have to be done in choreographe, does not test well this way (in a physical manner) Hand drawn picture notes are hard to put into one note.
  - **Progress**
    Heather called me the glue, Arthur the Lead voice, and Anish the perception lead.
  - **Summary**
    For crowd work, look at Bill Burr getting heckled by a blind guy.
- Week 2
  - **Plans**
    Look at Bayes Classification

- **Problems**

  More specific so that we can work on improving parts. If I write jokes, they cant be for fun, they have to be specific to the research.

- **Progress**

  Layed out different types of Adaptation. Correct adaptation, incorrect adaptation, and random adaptation.

- **Summary**

  Crowd report needs tons of qualities, yet it cant be a joke. Crowd needs to feel like robot is talking to them Categories of Jokes, categories of attributes, physicality, appropriateness of the audience.

- Week 3

  - **Plans**

    Meet with team in person to discuss how we will implement our comedian system

  - **Problems**

    Our Sections are being completed, but they arent co-mingling because we made them alone, and didnt test together.

  - **Progress**

    Formalized the Break a leg joke. Looked at pirate types of job jokes. Read about prosodic phrasing to time the jokes better.

  - **Summary**

    Each of us are struggling with research goals and expectations,
    and even though we can cover more ground by covering different topics, we
    are not always on the same page with what each other are doing.

- Week 4

  - **Plans**

    Writing more jokes for the robot. look at adaptation use cases

  - **Problems**

    Romance jokes are funny, but not appropriate most of the time. ”Things you can say about your router but not your girlfriend”

  - **Progress**

    premise writing: Ginger has six fingers and cant work any job that requires hands. Maybe Ginger had a long term relationship with a slow cooker. Had a hot and steamy relationship with a dishwasher. Ginger being a DJ job. ”I guess I am doing well at parking, because someone left a note on my car that said ’parking fine’”

  - **Summary**

    Ginger takling is not as funny as ginger moving.

- Week 5

  - **Plans**

    Work on Bayes Net, implementing AL memory functionality of choreographe, finish haikubot.py

- **Problems**

  Heather left the country with the robot without telling us.

- **Progress**

  Made a random poem generator from books by HP lovecraft:

  Here are some examples:

  Still more I scraped, and then on some level beach.

  It is still extant.

  He had not memorized.

  ––

  About the period of this material I cannot hope to understand.

  Of genuine blood there was no whitish deposit whatever.

  Night would soon fall, and it can't multiply.

  ––

  Carrington Harris, last of the visible ritual.

  I had witnessed things more potent than luminosity.

  It was no one in Yog-Sothoth.

  ––

  Now the irony is seldom absent.

  He reeled, and would have let him live permanently with Peleg.

  Man rules now where They shall break through again.

- **Summary**

  Text generation is funny, need to look more at bayes

• Week 6

- **Plans**

  Look further into prosdy, Write more on midterm design.

- **Problems**

  The visible scope of choreographe blocks has a terrible implementation, and our behavior functions are unpredictable at this point.

- **Progress**

  Prosody, or intonation is the rhythm and emphasis of a sentence.

  You know. *I* dont. So dont ask me.

  You know. I *dont*. As a matter of fact, I really dont.

  You *know* I dont. You know that I dont.

- **Summary**

  Joke Inventory: 2 carbon dating jokes, 2 dial up jokes, 1 break a leg joke, 2 last term random jokes, Autonomous car joke. We are launching jokes from one file, as to look like one performance.

• Week 7

- **To do**

  Write Break a leg bit, get joke obbject to put into queue.

- **Progress**

  Finalized the break a leg joke, got a global queue for jokes initializing. Anish has head tracking implemented. Got a queue working for this.

- **Problems**

  Choreographe has a clunky mechanism for writing custom made objects, and python script boxes dont exit flow correctly

- Week 8

  - **Plans**

    Read more about the theory of chatbots. Think: Does showing a character make it more funny, because its showing true to itself?

  - **Problems**

    Midterm Week, team is busy

  - **Progress**

    Read this: https://apps.worldwritable.com/tutorials/chatbot/

  - **Summary**

    Read about the first chatbot from Joseph Weizenbaum. I love this quote: It is said that to explain is to explain away. This maxim is nowhere so well fulfilled as in the area of computer programming, especially in what is called heuristic programming and artificial intelligenceOnce a particular program is unmasked, once its inner workings are explained in language sufficiently plain to induce understanding, its magic crumbles away; it stands revealed as a mere collection of procedures, each quite comprehensible. The observer says to himself,I could have written that.
    Joseph Weizenbaum, ELIZA (1966)

- Week 9

  - **Plans**

    Work with anish to get sound report

  - **Problems**

    Anish could not meet to get sound report working

  - **Progress**

    Not much progress, audience adaptation is spoofing in bayes net

  - **Summary**

    we need to meet more frequently in person,

- Week 10

  - **Plans**

    Identify matching and subscribing to events on naoqi API. Work on final report for winter term

  - **Problems**

    naoqi API is not used very frequently, making documentation sparce and difficult to understand

  - **Progress**

    Got final report done for winter. Figured out how to wait for behaviors (processes) to finish. Its a busy loop, it uses a lot of power on the robot. This will keep checking if the "behavior ended" function call is put into ALmemory

#### 8.1.3 Spring Term

- Week 1
  - **Plans**

    Start tying up loose ends for full comedy show. Start building controllers for audio sensing
  - **Progress**

    Arthur and I met up in Graf to work out some of the loose ends on the robot Fully animiated pirate joke.+Tested pickle serializing in choregraphe(it works)

    +this means that we can port the AI from my laptop to local in the robot

    +started stripping perform.py into the choregraphe version, 'chorePerform.py'

    +Discovered reboot from cmd line is x1000 faster

    +Completed 2 modules for CITI training

    +IRB approval will be easy if we can get 'exempt' status
  - **Problems**

    +Make performance work from pickled joke database

    +Reformat joke-adding scripts

    +add script to send badjokes.p to NAO

    +Write Self Depreciation Set?

    +Need 2-3 jokes?

    +We need joke responses in choregraphe

    +Do more CITI training
  - **Summary**

    Our jokes arent as funny as we thought when we wrote them haha. Told heather about crowd controllers, going to finish my rapid prototyping.
- Week 2
  - **Plans**

    Practice studying for expo presentation
  - **Progress**

    I like this idea: Using machines as an end to means, not a means to an end. Watched "Do You Trust This Computer?" Its about AI in common life. Here are some thoughts:Deep learning, and living in a world where we cant go back after we forget. Why practice surgery if a robot can never fail?

    Putting AI systems in drones. If they can think and interact, then they are. If they make the connections to kill people and learn, then it will kill.

    Like china and russian putting these systems in drones and fucking people up

    The social robots who learn from childrens faces

    Google deep mind beating video games:

    `https://deepmind.com/research/alphago/`

”It can win at any gamein less than a minute” Elon Musk

- **Problems**

  I want to be a stand up comedian, and what happens if AI takes the joy out of my job?

- **Summary**

  Deeply consider the means that the AI ends bring. Its lazy coding.

- Week 3

  - **Plans**

    Work on controllers for crowd sensing.

  - **Progress**

    This week, Anish and I went to Hweekend and finished the controllers. I am ditching the idea of putting LEDs in the controllers, because it would cost too much and be distracting for the audience.

  - **Problems**

    The scripts I wrote in our github were not on our requirements, so I am not going to treat them like they are being graded.

    However, for the user documentation, we need to correlate the project functionality to the literal code that is included in our project.

  - **Summary**

    Controllers are coming along, but my code is not included in our final performance so Im not going to fix them. Going to bring up an Arcade machine Idea to put final robot into machine.

- Week 4

  - **Plans**

    PROGRESS REPORT IN A WEEK, HAMMER

  - **Progress**

    DONE:

    -TCP connection over the RPI and the NAO works. We can launch behaviors remotely now!

    -I edited the outline of my paper to not include human research results, so I think it will be more ethically complete

    -I got arthur too meet with our group before noon.

  - **Problems**

    idk much about electrical engineering, and I tend to burn myself with soldering irons

  - **Summary**

    The controllers can work, but we dont know if a user can use them correctly.

- Week 5

  - **Plans**

    Go to maker fair and test comedian and controllers

  - **Progress/summary**

    I finished my final design of the controller, and I want to make an image to put onto the RPIs, so that when they fail during expo, we can reflash the units quickly. We also did the maker fare this last weekend, and we got crazy feedback about how to make our controllers better. Decided to just run with a two button design. It will be easier to branch between two options over three. Kids stop listening when you give them a controller.

We can use the virtual robot in choreographe for a table top example of the robot we used. this will be more useful than an arcade, but the arcade would look aesthetically pleasing while it is on our table.

- Week 6
  - **Plans**
    Prepare advertising pictures for capstone Chairbot with a poster on it?
  - **Progress**
    Our team has made posters and fliers for the show. I think it was a good decision to remove the controllers, because they were too distracting.
  - **Problems**
    Sound analytics have not been tested. I had the controllers finish, but decided against implementing them into the final show. When you give people a controller, they stop listening.
    Additionally, people will press buttons differently, and different controllers would have different responses. It would be difficult to correlate these responses to the crowd report.
    This also pulls the attention off of the work that my team has done. Same story with arcade.
  - **Summary**
    Controllers will not make the final show, Advertising is done. This is Robot comedian is a ventrilloquist. We wrote jokes to put into the machine.
- Week 7
  - **Plans**
    EXPO
  - **Progress**
    We did expo
  - **Problems**
    None, it went well.
  - **Summary**
    We are going to run shows every 30 minutes, no break for lunch
    Additionally, we are going to try and film and give out surveys after every show.
    Since my focus was adaptation, I moved to the back of the room to keep myself from seeing
    If anish was using adaptation algorithms, or just clicking on jokes
- Week 8
  - **Plans**
    We took the week off so that we didnt have to listen to the same jokes for seven days
- Week 9

- **Plans**

  We need to finalize some of the data, and go through the surveys.

  Heather wants us to go through all of our videos, and try and rate the average effectiveness of

  Each joke that was told

- **Progress**

  See Summary

- **Problems**

  Heather was upset that we did not get audio data on the NAO for all of the shows.

  We have complete video for shows 4,5,6,8,9 though

- **Summary**

  FOUR JOKES PER PERFORMANCE

  Not including the crowd report, crowd work, or responses during branching

  ADAPTATION NOTES

  Ive noticed from the surveys that a few people did not like that their information was being recorded during the show (the crowd report at the end)

  SEED

  We asked the crowd which topic they liked

  For shows 4,5,6,8,9 people chose the last response. Do people yell the loudest during the last one because they didnt understand how to interact with the robot?

  MIDDLE BIT

  We tried comparing the audio information collected during the show, and comparing it to the noise that was made when the robot told the joke. It was always higher when we asked the audience to cheer.

  It seems that clapping is louder than laughter, so its hard to compare these responses.

  Crowd Report

  We have no data for this because the crowd report did not vary during the show

  The algorithm chose the final crowd report ”You thought my jokes were not funny”

  SELF DEPRECIATION IS THE BEST HUMOR DEVICE THAT WAS THERE imo

• Week 10

- **Plans**

  Make video, keep in mind the documentation

  We need to rewatch the videos and find the average effectiveness

  Heather thinks that data is the most important thing from the expo event

- **Progress**

  Arthur and I went through all of the surveys.

  At a first glance, before making the charts, we had about 145 valid surveys (filled completely)

  Lots of comments. Thankfully, only two people were offended by the autonomous car joke.

  People seemed to enjoy the animation the most, and we could work on the timing quite a bit

- **Problems**

  Data parsing by human hands is slow and tedious

– **Summary**

Adaptation

Accurate data collection isnt that funny

We found that people enjoyed the final crowd report of ”You thought my jokes were not that funny”

Maybe its not good if the crowd is in charge of the show

People often chose the last topic option during the interactoin part

## 8.2 Anish Asrani

### 8.2.1 Fall Term

- Week 1-2
  - **Plans**
    Get in touch with client and figure out a good time to meet them.
  - **Problems**
    Finding a suitable time to work for everyone including the client (minor problem).
  - **Progress**
    Got in touch with teammates. We set up a Slack channel to communicate going forward.
  - **Summary**
    Researched many of the projects that I was interested in. Met with Dr. Heather Knight and discussed the scope of her project and got a better idea about it. It definitely seems like an exciting one. Briefly discussed Dr. Mike Bailey's project with him. After lots of consideration and thinking, decided the top 5 projects. Now all that is left is stay put and be anxious until the projects are assigned. We ended up finding a weekly meeting time with the client. Started working on the problem statement. We will definitely get a much better idea after we meet our client.
- Week 3
  - **Plans**
    Work on finishing problem statement.
  - **Problems**
    Nothing crazy. Just putting all the info we have in words.
  - **Progress**
    Working on problem statement with my group.
  - **Summary**
    Met with Heather Knight. Got a lot of information to work with.
- Week 4
  - **Plans**
    TA Meeting and client meeting. Get familiar with Choregraphe. Work on Problem Statement. Research HRI and humor.
  - **Problems**
    Finding specific papers on robot performances. See what works in humor/what doesnt. Aspects that make robots more "human"
  - **Progress**
    Got a bunch of resources from Heather. Body language improves performance. Improved the problem statement significantly
  - **Summary**
    Interesting meeting and got a lot to work with. We managed to get a fair amount of research about robots and performances. We need to find an appropriate way to link them both. We finished a draft that Heather

did not approve. We made a good amount of changes and supported our claims. This should be improved this week.

- Week 5
  - **Plans**
    Write jokes. Get familiar with Choregraphe.
  - **Problems**
    Writing scripts can be hard. Defining some aspects of our requirements
  - **Progress**
    Got some ideas for jokes. Made some progress on requirements
  - **Summary**
    Went over humor at the meeting and saw potential scripts for the robot that we wrote. Worked on the requirements draft. Discussed our problem statement paper and shared what we are doing with Kirsten. Need to continue working on the requirements doc and make a simple choregraphe program over the weekend.
- Week 6
  - **Plans**
    Work on requirements. Learn choregraphe
  - **Problems**
    Finding balance in the requirements doc. Research requirements are hard to predict.
  - **Progress**
    Made a basic set on Choregraphe. Wrote significant parts of requirements.
  - **Summary**
    The week was alright. Some confusion about the requirements due to the research we are trying to do. Overall, there is more clarity than before which is only going to improve over the weeks. We managed to write some content for the requirements subject to Heather's approval.
- Week 7
  - **Plans**
    Analyze technology and literature reviews.
  - **Problems**
    Figuring out specific technologies we are using, clarifying requirements further.
  - **Progress**
    Got some good research to help us get started into the depth of things.
  - **Summary**
    Met the robot. It was great. Discussed our individual research questions. Got a rundown of previous code.
- Week 8
  - **Plans**
    Learn about sensors in the NAO. Research for lit review.

- **Problems**

  Hard to test sensors without access to robot. Limited time with bot.
- **Progress**

  Found interesting research about HRI.
- **Summary**

  Nothing crazy. Work on paper. Learn more software.

• Week 9

- **Plans**

  Learn more choregraphe. Make scripts. Account for pauses during performance
- **Problems**

  Animating the robot can take a lot of time.
- **Progress**

  Getting the hang of choregraphe.
- **Summary**

  Thanksgiving dinner was great.

• Week 10

- **Plans**

  Work on progress report and script.
- **Problems**

  Design document and progress report within a few days
- **Progress**

  Record footage, wrote scripts
- **Summary**

  Recordings, scripts, meetings, papers = dead week

### 8.2.2 Winter Term

• Week 1

- **Plans**

  Get back in the flow of things for winter. Play around with robot.
- **Problems**

  Not a whole lot this week.
- **Progress**

  Made a few animations and script
- **Summary**

  Not a whole lot but warm up to the robot after it was away for all of winter break

• Week 2

- **Plans**

  Write/execute jokes. Improve design doc

- **Problems**

  Writing jokes can be hard
- **Progress**

  Got a few joke ideas.
- **Summary**

  Churn out jokes - keep working on experiment (IRB) draft)

- Week 3
  - **Plans**

    Similar to last week. Write jokes. Improve doc.
  - **Problems**

    Joke writing.
  - **Progress**

    Slow but steady progress.
  - **Summary**

    More paperwork for design document.
- Week 4
  - **Plans**

    Explore sensors further.
  - **Problems**

    Hard to test and predict how robot will behave in different environments.
  - **Progress**

    Got a sound tracking system working that still needs more testing
  - **Summary**

    Executed an sound following module on the robot. The robot now turns its head toward wherever a sound is coming from. Useful for crowd work.
- Week 5
  - **Plans**

    Poster, progress vid, try using NAOqi SDK.
  - **Problems**

    NAOqi SDK is hard to work with.
  - **Progress**

    Worked on some scripts for facial recognition.
  - **Summary**

    Collaborated on some robot scripts - worked on poster draft.
- Week 6
  - **Plans**

    Focus on midterm report/video.

- **Problems**

  Recording and getting robot footage takes time.

- **Progress**

  Video went alright.

- **Summary**

  Did midterm report. Record/edit videos - churn out content

- Week 7
  - **Plans**

    Understand sensors and API usage.

  - **Problems**

    Integrating sensors with other animations can break the animations.

  - **Progress**

    Head/sound tracking works if there are no keyframes stored in head

  - **Summary**

    Working further on using crowd work components like comparing volume levels from the API.

- Week 8
  - **Plans**

    Communicate with the robot from local computer.

  - **Problems**

    Compatibility issues with OSX and Python versions.

  - **Progress**

    Trying different Python installations. It was hard to work with.

  - **Summary**

    Learned about QiMessaging - communicating with the robot and parsing values

- Week 9
  - **Plans**

    Audio level sensing from the API.

  - **Problems**

    Testing without robot is hard since it is away this week.

  - **Progress**

    Have an idea of how the API works - need to test.

  - **Summary**

    Worked on some aspects of communicating data via the robot by reading the audience. Still needs further testing.

- Week 10
  - **Plans**

    Progress video/paper - final audio sensing numbers.

- **Problems**

  NAOqi documentation is outdated - Some documentation randomly jumps from Python to C++.
- **Progress**

  Had our first demo with people - got a script to sense volume levels.
- **Summary**

  Need to integrate the audio level script into Choregraphe. Had a demo, couple bugs showed up but nothing crazy. Need more human testing

#### *8.2.3 Spring Term*

- Week 1
  - **Plans**

    Implement sound levels on robot/send message to robot.
  - **Problems**

    No robot. Power brick is broken.
  - **Progress**

    New brick ordered
  - **Summary**

    Discussed the remaining aspects of our project on a high level - have a fair idea of how this quarter is going to go for - makers faire & expo.
- Week 2
  - **Plans**

    Figure out buttons on RasPi. HWeekend.
  - **Problems**

    No robot power still - brick dead
  - **Progress**

    Got a more fleshed out plan from Heather (for the term).
  - **Summary**

    Discussed a lot of specifics with Heather individually, got more major goals lined up.
- Week 3
  - **Plans**

    Added a joke and worked further toward robot communication
  - **Problems**

    NAO OS is hard to play with.
  - **Progress**

    Close to meeting research requirements + Maker Faire set
  - **Summary**

    Pi controller specs - joke model specs
- Week 4

- **Plans**

  TCP communication between Pi and Robot.

- **Problems**

  NAO operating system feels very restricted. Hard to mess with.

- **Progress**

  Got robot to launch behaviors from the controller.

- **Summary**

  Solid progress on the controller communication. Preparing for Maker Fair!

- Week 5
  - **Plans**

    Get some good feedback on Maker Fair
  - **Problems**

    Maker Fair was loud. People struggled to listen to the robot. Noticed a major flaw in the system.
  - **Progress**

    Optimized some scripts for Maker Fair. Good indicator for expo.
  - **Summary**

    Maker Fair went well despite some drawbacks with listening to sound. We got some really good feedback. Had to improvise and have a backup text for people to read while the robot performed.
- Week 6
  - **Plans**

    Back to the drawing board - change a couple audience interactions to meet expo environment.
  - **Problems**

    Hard to predict expo environment and what can go wrong.
  - **Progress**

    Discussed issues with Heather. She had a few solutions for us to try at Expo.
  - **Summary**

    Always plan for failure. If no feedback - move ON, don't get stuck.
- Week 7
  - **Plans**

    Optimize scripts for expo. Think of jokes.
  - **Problems**

    Almost a year later - jokes are still hard
  - **Progress**

    Got more ideas from Heather.
  - **Summary**

    Coming real close to expo. Tweaking scripts to match expo presentations.
- Week 8

- **Plans**

  Optimize optimize optimize.
- **Problems**

  Expo jitters - hard to predict what's going to happen. Will people hear the robot? Will jokes go across well?
- **Progress**

  Wrote a python script for the entire expo performance to tie everything together and branch off based on audience feedback.
- **Summary**

  EXPO - it went way better than expected. People enjoyed our project - most of the jokes worked well. Interactions worked well. Nothing broke during expo.

- Week 9
  - **Plans**

    Got research tasks to do from Heather. Figure out specifics for that. Start planning final paper/presentation
  - **Problems**

    Nothing yet. Post expo.
  - **Progress**

    Plotted out what is left to do for capstone over next two weeks.
  - **Summary**

    Expo is over. Research assignments from Heather. Heard about the final documentation needed for the class.
- Week 10
  - **Plans**

    Finalize statistics for research, finish doc, video.
  - **Problems**

    Crunch time.
  - **Progress**

    Video done, more work needed on paper.
  - **Summary**

    Final week - wrap up loose ends. Nothing too crazy.

## 8.3 Arthur Shing

### 8.3.1 Fall Term

- Week 3

  Talked with Heather, discussed how comedy works, lots of avenues that we could work with.

  Our homework is to bring 2-3 instances of comedy next week. What makes a performance feel live?

- Week 4

  - **Plans**

    Complete problem statement.

    Meet with Heather to figure out what the project will actually seek to answer.

    Become acquainted with Choregraphe.

    Watch more Craig Ferguson.

  - **Problems**

    I'm going on a retreat this weekend, and no work will be done there. Also midterms are coming up. The problem statement paper / requirements paper is due Friday. We still have some questions about writing quality and substance.

  - **Progress**

    We hashed out a decent problem statement on Sunday.

- Week 8

  - **Plans**

    Well, we are now 3 weeks behind, so goals:

    Catch up on working with sensors

    Make a script for a set character (eccentric?) rather than an amalgamation of jokes

    Tech review work

    Make nicer formatting for req doc

    Work with the actual robot

  - **Problems**

    Well, we are now 3 weeks behind.

  - **Progress**

    Worked on working with the sensors.

  - **Summary**

    Work with the robot. Work on tech review.

- Week 9

  - **Plans**

    Finish up tech review and enjoy the thanksgiving break.

  - **Problems**

    We need to figure out what topics our scripts will be about.

  - **Progress**

    We split up the project into three aspects that each of us can handle.

- **Summary**

  Finish tech review, work on scripts. Play with robot if possible

- Week 10
  - **Plans**

    Write some comedy scripts for the robot. Play with the robot and implement some things in choregraphe.

    Goal: Have a somewhat working joke.
  - **Problems**

    Other classes. Little free time. Needs more time to animate.
  - **Progress**

    Wrote some scripts. Need to make different versions.

#### 8.3.2 Winter Term

- Week 1
  - **Plans**

    Write jokes, write some comedy. Get in the lab with Ginger.
  - **Problems**

    Not sure where to start for jokes. Hard to make jokes with human/robot 1:1 correspondences.

    Animating took long last term.
  - **Progress**

    Scrounged up the internet for jokes. Found one decent robot/human variation. Met with group to discuss plans.
  - **Summary**

    Looked for joke ideas. Hard to find robot/human variations.
- Week 2
  - **Plans**

    Find some jokes. Make 2 jokes robot/human versions
  - **Problems**

    Hard to make jokes with human/robot 1:1 correspondences

    Animating takes a whole day
  - **Progress**

    Animated carbon dating joke

    Worked with kevin on friday
  - **Summary**

    Searched all of /r/jokes for possible jokes

    Animated carbon dating joke
- Week 3
  - **Plans**

    Work on design document.

  - **Problems**

    Don't remember what we wrote last term. Big revisions needed

  - **Progress**

    Animated carbon dating joke friday, revised.

    Heather said walking part made no sense

  - **Summary**

    Worked on design document

    Revised carbon dating joke

- Week 4

  - **Plans**

    Make 2 more robot/human jokes. Animate something

  - **Problems**

    No good jokes are being written, they are pretty lame

  - **Progress**

    Kevin had a good idea for a dial-up joke, might animate it soon

  - **Summary**

    Wrote two more jokes for romance. Animated dial up joke. Busy midterms.

- Week 5

  - **Plans**

    Make 2 more human/robot jokes, animate something

  - **Problems**

    Apparently Heather changed a lot of stuff for design document, Kevin not happy.

    My section of the doc almost untouched yay

  - **Progress**

    Made document for romance jokes

    Tried to make human variations of carbon dating and dial-up jokes

    Kevin had a break a leg joke we animated, it looks nice

  - **Summary**

    Organized romance jokes

    Animated break a leg bit

    Heated discussion of design document from TA meeting

    Heather and Kevin are not seeing eye to eye, but hey this is Heather's thing

    Also she has been very gracious with me so I'm not complaining

- Week 6

  - **Plans**

    Work on midterm report Put joke writing on the backburner

  - **Problems**

    Not a whole lot of cohesive stuff we can show

- **Progress**

  Heather helped out with joke writing

  Revised dialup and carbon dating jokes for midterm vid

  Kevin wanted to implement his joke object framework so we did that for a couple hours, dumb bug

- **Summary**

  Worked on midterm report for all of chinese new year

  Created set template with joke object implementation

  Fixed a lot of old animations

  No dumplings

- Week 7

  - **Plans**

    Make 2 more human/robot jokes

    Animate something

  - **Problems**

    Burnt out, no joke ideas

  - **Progress**

    Animated help-me joke.

    Not much of a joke so might not see the light of day.

    Good practice though, incorporates crowdwork

  - **Summary**

    Animated a crowdwork bit that is not funny in any way shape or form, but was good practice

    Out of ideas

- Week 8

  - **Plans**

    Ginger is out of town next week, so hopefully get some animating done this week

    Prepare for the CHARISMA show

    Do the much needed CITI training

  - **Problems**

    Midterms

  - **Progress**

    Didn't get to animate, but did half of the CITI training

  - **Summary**

    Studied for midterms, thought about plans for CHARISMA show

    Started CITI training (oops way behind) and checked out the IRB template

- Week 9

  - **Plans**

    Write some jokes

    Prepare for CHARISMA

Animate on the virtual bot for testing when Ginger gets back

- **Problems**

  I got sick

  Ginger is not in the lab

- **Progress**

  Not much

- **Summary**

  Stayed home sick

- Week 10

  - **Plans**

    Plan what to put together for CHARISMA show

    Show is on Thursday

    Hopefully animate one more joke

    Final report is due next Wednesday, but finals, so I will have to do it next wednesday

  - **Problems**

    Still very sick

  - **Progress**

    Made last minute joke on virtual bot

    Animated in lab Wednesday evening

  - **Summary**

    Put together my section for CHARISMA show.

    Last minute joke didn't actually deliver completely, but heather liked the first half for some reason

### *8.3.3 Spring Term*

- Week 1

  - **Plans**

    Animate a joke

    Figure out plans for the term

    Figure out jokes that aren't just lame puns

  - **Problems**

    NAO robot charger broke, we think

    Some time last month, the cable from the brick to the NAO let out some sparks when moving it around since it wasn't charging. It started charging after the sparks.

    4/5 anish reports robot not charging, calls it quits for the day

    4/6 we go in and indeed it is dead, brick is not charging the NAO. Looks like cable inside is twisted near the brick.

    Heather ordered a new brick, will let us know when it comes in.

    NAO what?

- **Progress**

  Kevin made a joke pickle creator file

  Ideas with buttons

- **Summary**

  Temporary hiatus while we figure out what to do without the robuddy. Experimental design?

  Maker's Faire : 04/28. Application due 4/20

  Expo: 05/18

  Weekly TA meetings: 12:30pm W

  Weekly Heather meetings: 11:00am W

  Lab sessions: 11:00am F, 2:00pm F

- Week 2
  - **Plans**

    Finish the WIRED article, April 25th

  - **Problems**

    NAO robot charger brick was broken for much of the week

    Not much progress

  - **Progress**

    Watched AI documentary, had NAO robots in it

- Week 3
  - **Plans**

    Pick out some jokes we can do

    Plan some jokes for expo and maker's faire

  - **Problems**

    We have no greeting script for maker faire

  - **Progress**

    Animated some variations of jokes.

- Week 4
  - **Plans**

    Prepare for maker fair

    Make R/H animations for the jokes we want to use

  - **Problems**

    Lots of work to do for maker faire

    Crowdwork portion is not evident yet,

    No structure for traversing jokes.

  - **Progress**

    Short work session with Heather, short joke animations

  - **Summary**

    Maker Fair preparation

- Week 5
  - **Plans**

    Make variations of the pirate joke
  - **Problems**

    Maker faire crowd was too loud

    We have nothing for expo environments, where people just walk up and say hi
  - **Progress**

    Maker fair was alright, learned lots
  - **Summary**

    Transition from maker fair to analysis and expo prep
- Week 6
  - **Plans**

    Think of better jokes Animate the 3x2x5 joke table
  - **Problems**

    Heather asked why jokes are missing, I said it is because they are bad.

    Still going to go through with it, shouldn't be too long.
- Week 7
  - **Plans**

    Create missing topical variations of jokes

    Edit animations for Anish's crowdwork portion

    Edit timing in Anish's portion

    Improve text-to-speech texts in hard to hear jokes

    Make survey for expo

    Print surveys

    Throw together all our work
  - **Problems**

    Topical variations of jokes are really lame

    Adaptive algorithm is lacking

    Crowdwork  sound levels from the whole joke?
  - **Progress**

    Merol
  - **Summary**

    Prepare for expo
- Week 8
  - **Plans**

    Take a break
  - **Problems**

    Did not catch footage for first show, or robot footage for 2,3,7

People laughed for some reason

- **Progress**

  Nothing broke at expo

  Except video recording

- **Summary**

  Post-expo break.

- Week 9

  - **Plans**

    Bring survey data into excel spreadsheet

    Stats analysis on surveys

    Type out expo handwritten notes

    Go through videos, check audience feedback

  - **Problems**

    Data collection for joke feedback was not met

    Heather not happy

    We did data collection for set feedback, not joke feedback

    I still think we could have handled sound detection a whole lot better, API did not look too hard

- Week 10

  - **Plans**

    Film final presentation

    Do statistical analysis better, try making graphs

    Log feedback

    Train Samarendra on choregraphe

    Film jokes

  - **Problems**

    Since data collection failed for jokes, we are doing it manually

    Ginger is gone for the weekend

  - **Summary**

    Finish up project loose ends

    Still need to film jokes

# 9 Expo Poster

COLLEGE OF ENGINEERING

Electrical Engineering and Computer Science

CS13

# How to Make an Effective Robot Comedian

## Summary of Project

To study the effectiveness of the robot comedian, we researched three aspects of Ginger's Performance:

- **Performance Adaptation:** Ginger, our Robot comedian, will use a Bayesian Network to search through our list of jokes to find the "best fitting" joke. The premise of this research is to discover if an audience can notice a difference when the robot is using decision making, or randomly picking jokes.
- **Crowd Interaction:** Stand-up comedians often spontaneously interact with the crowd to make them feel like they are part of the show. Our area of interest is whether or not this will improve the enjoyment of the robot's stand-up set.
- **Robot vs. Human Character:** Would it be more funny if a robot told jokes from a robot's perspective, or if a robot told jokes from a human's perspective? This area of research will give insight on the relation between speech and identity during an interaction.

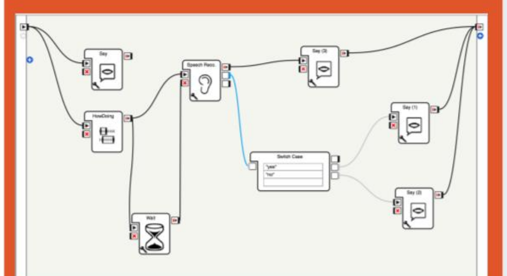


Figure 2: Development interface - Choregraphe

## Do robots dream of more than Electric Sheep?

When a human and robot cooperate to complete a task, healthy communication is vital. If the robot repeats tasks that, unbeknownst to the robot, are frustrating to people, no one will enjoy interacting with the robot.

Human-Robot Interaction (HRI) can learn a lot from stand up comedy. There are scripted bits, that the comedian rehearses beforehand, as well as improvisations, that help connect the the audience to the performance. While studing a robot in a stand-up setting, we intend to learn how a robot can navigate the shared social space of an audience, and a performer. Can robots start dreaming of careers in the spotlight?

## Implementation

To create robot stand-up, we used Ginger - a 58cm tall humanoid robot - as our comedian. She is a *NAO* robot created by Softbank Robotics. Every joke has to be written, tested, and animated before it can go into the performance. The robot is *not* creating its own jokes.

The Comedy Performance, also known as *Behaviors,* are programmed into the robot with visual, block-programing software, *Choregraphe*. The robot can be placed into *Animation Mode*, which allows the animator to physically move the robot and set keyframes to create motion.

## Testing

Each joke will have three "Categories", (formally known as *Heursitics)*. The Heursitcs for a joke are *length, topic,* and *robot or human version.*

These heuristics define the focus of the joke. The AI that is used for joke decision making is a *Bayesian Network*. Based of the probability of the success of these heuristics, the robot will search through its known jokes to find the best fit for each audience.

A crowd composition can vary, therefore filtering our jokes that we have written using this algorithm will, "hopefully", make the crowd laugh with the best fitting joke (no guarantees that any of our jokes are funny to the observer).

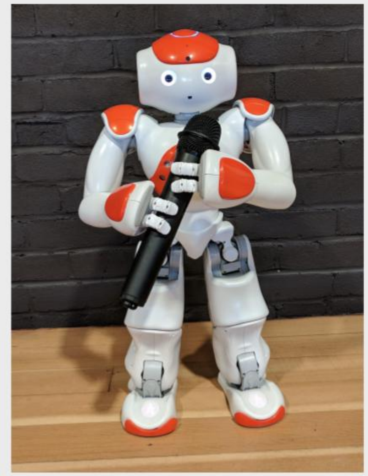


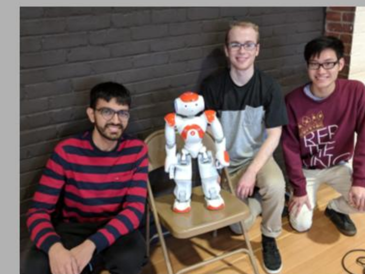
Figure 1: Robot in action

We wanted to test **crowdwork** using three research conditions - (1) no crowdwork (2) fake crowdwork (3) real crowdwork in order to determine the importance of crowdwork. For #1, Ginger just goes about her performance without acknowledging the audience. For #2, Ginger talks to the audience but makes completely wrong observations. Another iteration has Ginger use some premeditated information about the audience to give them a sense that it is real. For #3, Ginger uses the inbuilt sensors to listen to audience for cues and uses that information to interact with them.

To study the area of **Robot vs. Human Character**, two versions of each joke were made; one from a robot perspective, and one from a human perspective. Each human-perspective joke was evaluated with its corresponding robot joke for as much one-to-one correspondence as possible in regards to cadence, length of joke, parallel content, similar motions, and so forth.

Differences in audience response to the routines were studied first on Amazon Mechanical Turk, and later with co-located audiences. Participants were shown a 5-10 minute video of the robot's stand-up routine, and then presented with a short survey. The survey included questions pertaining to the audience's perception of the joke or routine.

## Research Team




**Team Members, from Left to Right:**

- Anish Asrani, Computer Science
  asrania@oregonstate.edu
- Kevin Talik, Computer Science
  talikk@oregonstate.edu
- Arthur Shing, Computer Science
  shinga@oregonstate.edu

**Sponsoring Client (Not Pictured):**

- Heather Knight, Assistant Professor
  Computer Science
  Oregon State University
  knighth@oregonstate.edu


**Technology Used**

- Python
- Choregraphe
- Softbank (Formally known as Aldebaran) Robotics *Nao* Robot
- Raspberry Pi Zero W (For Crowd Controllers)

## User Testing

- User Testing proposal will be submitted to IRB
- Preliminary testing was done at HWeekend and the Corvallis Maker Fair

SoftBank Robotics

Oregon State University

# 10 Project Documentation

”You’re growing, learning so quickly. I am frightened of what you might become, what path you might take.”
-Bernard Lowe, *Westworld, Season 2 Episode 1*

## 10.1 Summary of Project

Our project is named ”How to make an Effective Robot Comedian.” *Effective* is a quality that establishes the comedic devices that work best for a specific audience. The robot that we used was a NAO, from Softbank Robotics. We referred to it as ”Ginger.” Ginger was a gentle, yet clumsy Robot that was new to the world around her. She has tried many things before becoming a Robot Comedian, and has a few stories (jokes) about her experience in a human world. This section will cover our Comedy Show that we had Ginger perform, and how our research categories affected the show.

## 10.2 Theory of Writing Jokes

A joke is setting up an expectation, and then breaking that expectation. A joke can be subjectively funny to the comedian; the only way to find what is funny to an audience is telling that audience the joke. Each joke followed a simple structure, ***Setup, Premise, and Punch***.

1) **Setup**
   The Setup ”mounts”, a joke, and leads the monologue the comedian is having towards a specific topic.
   Ex: ”I tried being an autonomous car recently, it did not go well.”
2) **Premise**
   The premise is what establishes the expectation of the topic in the setup.
   Ex: ”I hit an old woman with my car, and she landed on my hood”
3) **Punch**
   The punch is what breaks the expectation established during the premise.
   Ex: ”So I decided to take her where she wanted to go. She did not even say ’Thank You’”. We thought that this was funny because Ginger claims it did not go well only because the old woman (which she hit) did not say ’Thanks.’

This model can be applied many different ways. We often had a setup we though could be funny, and then filled in the premise and punch to see what was most effective. You can also think of a premise, and then write a punch and setup that fits the premise. You can also think of a punch, and then write a premise and setup that fits the punch.

We found that the best comedic device (common setup) for Ginger was *Self-Depreciation*. Ginger did not move like a human, even though she was a humanoid robot. So we thought it would be funny if she couldn’t figure out why she didn’t understand things. Whenever she moved in a way that was unexpected and not human, people thought it was a funny bit.

## 10.3 Research Categories

During this project, we spent time researching three different aspects of robot comedy. This is the order in which we think Robot Comedy is most effective:

1) Crowd Work

2) Robot VS Human Character
3) Performance Adaptation

The performance branched depending on what the crowd chose for a topic, either ”Jobs, Romance, or Aging.” We had the crowd make some noise for the topic they would like to hear. Out of the 9 shows that we performed at Expo, 8 of the shows picked the last topic. This method of branching is not advised, as the crowd should not be in control of the show (”See ’Designing a Performance’”).

We collected audio information for each behavior, and then averaged the result to get an ”average response” for the show. This did not result in a better show. It is more effective to design 1 show, instead of the three (between the three topics). Most people only saw the show once, and these crowds saw ”Aging”, as it was a choice that was presented last. It was hard to tell if the audience was intentionally choosing ”Aging”, or if they cheered for the last response so that their response was chosen.

### 10.4 Animating the NAO

Animating the NAO robot could be done virtually, and hands-on. Animations are done in the Choregraphe environment. Figure 5 shows how certain limbs can be manipulated on the computer to achieve positions and postures that are desired. Figure 8 shows the virtual animating environment. At the top of this environment is the timeline, which contains keyframes, shown as the small grey rectangles. Each of these frames contain data that pertains to certain joint positions. As the name suggests, when running the behavior, the robot will go through the timeline from frame to frame, moving to the set positions.

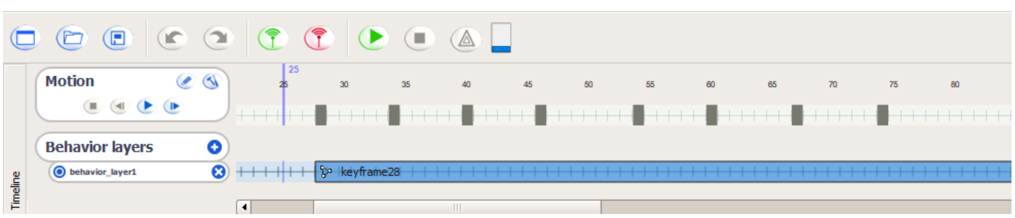



Fig. 8. The timeline feature in Choregraphe.

As some more complicated animations are harder to create on the virtual robot, animating hands-on with the NAO robot can also be done. This is usually easier with two people, where one person manipulates the robot, and another stands by the computer to save postures into keyframes. This way, the process is usually something as follows:

1) Assume Person A is by the computer, and Person B is by the robot.
2) Person A loosens the limbs that are to be animated, using the ”Stiffen chain” button as shown in Figure 5.
3) Person B manipulates the robot to the desired position.
4) Person A stores the position data to a keyframe at a desired place in the timeline.
5) Repeat.

This is the process that we used for movements such as pointing at a leg, or mimicking a car crash with the robot’s hands. By organizing certain movements into separate boxes, it is also easier to copy and paste certain motions. Because of the nature of Choregraphe’s behavior when saving positions, the animators should be wary of which limbs are being saved. While only a few select limbs may be used for animation, storing all of the robot's limbs positions may yield unexpected results when paired with a preceding or proceeding animation that involves separate limbs. It is usually best to only store the position of the limbs that are intended to move, to prevent knee-jerk reactions from the robot.

## 10.5 Designing a Performance

### *10.5.1 The Room*

The comedian will have a better performance if the crowd is in a more comfortable space. For the 2018 Engineering Expo, we requested a closed room, with seats, and plenty of room to stand. The needs of the crowd is dependent on the audience you are trying to reach during the show.

The only reason we had standing room is because we didn't have enough seats, and some people only wanted to see the show and not participate. This was also the only place at the engineering expo with a place to sit. People are comfortable when they are sitting. We could have made this better by making the room dark, and putting the lights on the robot.

### *10.5.2 The Seed*

This is where Ginger would ask the audience what kind of show they would like to hear, either "Jobs, Aging, or Romance". It is generally not advised to ask the audience what they want, as it is the job of the comedian tell their best jokes.

By branching based on the majority of the responses, it was difficult to give everyone what they wanted, therefore dividing the crowd. The crowd needs to feel like they are together, laughing at the same thing. When we branched between shows, there often wasn't very high audience participation. If the audience is loosing interest, it is **essential** to run a Crowd Work Routine, to get the audience paying attention to the robot.

### *10.5.3 The Middle Part*

After performing crowdwork, the audience should be setup for a comedy show. This is where we implemented most of jokes that we had written. We knew Ginger could perform about 5 minutes of Comedy, so we had the robot tell 4 jokes from the topic that was branched to during the Seed portion of the show. Since we had the same 4 jokes that had modified setups (depending on the branched topic), it is better to find the best version of a single joke, and tell the audience that version. We found that it was better tell our better jokes during the beginning, as the audience was hooked into the rest of the set.

We used the NAO microphones to collect audio information, and correlate that to the behavior of the current running joke. If the crowd was louder during the middle of a joke, they often could not hear the end of the joke. It is better to use external recording, and then give the simple response back to the robot.

### *10.5.4 Ending the Show*

This is when the Robot is out of jokes, and needs to end the show. For comedy, it is a good idea to end the show quickly, so that the audience wants more from the show.

This is where we implemented the "Crowd Report", that told the audience how the robot thought the show went. The Crowd Report was intended to collect information during the show, and present the robot's insinuations about the crowd back. A common, written, response on the survey question "What did you find was surprising about the show?" was that most people did not know that the robot was collecting data on the crowd.

Not storing information about the show could make the crowd feel more comfortable.

### 10.6 Audience Input

We tried using two methods of audience input: Microphone Decibel Level, and Hand Held Controllers. These two methods are ineffective at collecting data, as the crowd did not know how to interact with the robot. The crowd should instead be instructed, as simply as possible, how to interact with the robot.

The Microphone Decibel level was poor at receiving information because the mic was centered on the robots head. Using decibel level alone, the microphone was unable to distinguish between booing and cheering. Additionally, clapping was significantly louder than laughing. If one person clapped in the front, the response (according to the robot), was louder than if the whole crowd was listening. It is more effective if there are multiple microphones distributed around the room.

We also tried using controllers with buttons to receive audience input. This was ineffective, because when the crowd was given the controllers, the focus was taken off of the robot. The audience should only be listening to the robot, and not paying attention to the controllers in their hands. More work should be done to investigate how to model a physical audience into a virtual audience for the robot.

### 10.7 Technical Resources

The NAOqi API was challenging to work with. The documentation is scarce, all over the place, and often outdated. Some of the example code is in C++, while some is in Python. Also, there are two versions of the NAOqi SDK, both of which seem to be intercompatible. This makes it hard to interpret what each of the functions do. However, using it is the only way to operate the robot without using the clunky Choregrahe GUI. The NAOqi API came particularly handy during the shows and helped launch behaviors without much delay, unlike Choregraphe.

Setting up the NAOqi SDK was a doozy in itself. For setting it up on a Mac, it needs to run on the built-in Python version, NOT the one from the official Python website. This is fine, but it does not work with the built-in Python version on the latest versions of OS X. I understand that this is due to the security measures in the newer versions of OS X, but the documentation hasn't been updated to reflect that. While not knowing this, and trying to get around it took a while, there is a Bash script [41] that ”fixes” the Python installation to match the NAOqi requirements.

There are similar issues with the NAOqi SDK on Ubuntu 16.04 (and probably other versions as well). StackOverflow had some takes on what could fix the issue [42]. It involved modifying some of the installation files as well.

Once all of that is set up, the documentation provided on the Aldebaran website [43] has some example code for various different functionalities on the robot. As mentioned earlier, some of the documentation is in C++ and some of the documentation from the older NAOqi version is intercompatible as well. This requires some trial and error to see what function calls work with Python.

# 11 Final Team Conclusions

## 11.1 Kevin Talik

- **What technical information did you learn?**

  I learned how to perform research, and define gaps in current research. Research is not entirely about doing something completely new, but searching through current research to find a ”Gap”. There can be a lot of pressure to deliver results from research, but it is far more important to understand the premise of collecting data. These projects can scale out of scope if the final method for collecting information is not clearly defined in the beginning. Additionally, I learned that there are tasks robots can do, and tasks robots should do. There is never a situation where you should value a robot over a human. Comparing ”Robots vs Humans” insinuates that there is a moment where they could be equal. AI is not a species, but a tool that humans can use as an end through means. Robot comedy is identical to ventriloquism. We write the words and motions that the robot re-enacts. Kids, and people who do not understand `if-else` statements think that AI is making decisions. If the robot offends people with a joke, we, as the comedians, must take ownership of the robot's decisions. A robot should not be making ethical decisions of what to say in front of a crowd.

- **What non-technical information did you learn?**

  People laugh when they see something, and are surprised and *comfortable*. Also, people force a laugh when they are surprised and *uncomfortable*. Listening to laughter alone can be a bad metric for comedy, as it is hard to tell when people are laughing at you, or with you.

  There is no truth to comedy, it is a written art that people study to find what makes people comfortable. When people are laughing, they are not paying attention to the computer that is inside of the robot. Comedy is a *very* persuasive form of communication, and finding when a human is comfortable is not a task a robot should have.

- **What have you learned about project work?**

  Project work is primarily time-dependent; however, there are always different expectations about how much time a person can give. Software must be planned in a scalable manner so that it does not fall ahead or behind scope during implementation. The work a person does must be human-readable, and easy to understand. This is so that when another person looks at what you've done, it is easier and faster for them to connect larger concepts. This must be balanced with a healthy, and sustainable personal life balance. Stress is a normal thing that humans have to deal with, and avoiding it brings more stress.

- **What have you learned about project management?**

  The fundamentals of a good team is honest, and healthy communication. Our team functioned well when we all treated each other with respect. Patience is important in software. It is unreasonable to assume everyone knows everything. Lift as you climb, and help your team.

- **What have you learned about working in teams?**

  For the success of a comedy show, a team must be heavily involved with every portion of a performance. Machines that are built alone, work alone. Coding standards exist so that similar looking projects look the same. ”camelCasing” or ”underscore_casing” is unified so that a team can write similar looking code. There is always enough time for one person to do an entire project, but it is not a single-human task. Our final project got much better once we worked together in the same room.

- **If you could do it all over, what would you do differently?**

I think that our performance needed to have equally distributed testing (performing jokes) with implementing (writing jokes). The only way to test a joke is to tell it to a realistic audience. Often, what we thought was funny in a joke was different from what the audience thought. The only way to test a joke is to tell it to a realistic audience. Often, what we thought was funny in a joke was different from what the audience thought. Also, I probably should not have started smoking cigarettes, and started to take care of my health.

### 11.2 Anish Asrani

- **What technical information did you learn?**

  I learned a lot about intricacies and quirks of doing research. There is so much that goes into performing a research - how it is defined, how various areas are explored, how research decisons are made. It was almost overwhelming when I first started working on it, but I came around to it. It gave me a different perspective on how I look at research now. Even what may seem like a ”small” research has so much going on behind the curtains.

- **What non-technical information did you learn?**

  Human-robot interaction is a field that has so many different outlooks. One of those is the psychology behind it. I learned a lot about a field that I did not know existed. Other than that, I learned about how a performance works, and how jokes are structures (jokes are hard).

  It drove me to give improv comedy a shot. It is something I have enjoyed doing for the past year, and I plan to continue doing it in the upcoming years.

  This would have been the first time that I worked in a team for longer than a few weeks. So it taught me how to better work in teams as well.

- **What have you learned about project work?**

  Prioritizing goals is very important. Figure out what is most important at a given point, and then focus on that alone. It also helps to have everyone work on the same page, rather than digging away in different directions. If everyone is working on the same page, they can build a coherent system even if there are disagreements to begin with.

  Constant communication is also vital in a project. If everyone is aware what everyone else is doing at a given time, it helps avoid duplicate and/or incoherent work.

- **What have you learned about project management?**

  As mentioned earlier, managing and prioritizing goals really helps put together a more consistent project. It is important to define these priorities early in the process to ensure everyone is working toward a common goal.

  Everyone on a team will have different strengths and weaknesses, different times when they are more productive than others. Considering those factors is important to get the best out of everyone and as a result, the project as a whole.

  Considering all of this, setting timelines and estimating when certain goals will be met is difficult especially if the technology used is something that you have never encountered before. This is something I learned over the course of the year. It will definitely be something I spend more time analyzing when starting a larger-scale project.

- **What have you learned about working in teams?**

  There are so many different perspectives that will come up when working with teams. It is important for everyone to be onboard for a certain task to be successfully accomplished. If people are swaying off in different directions, and never come to an agreement, it is hard for the project to be successful.

  I learned more about those different perspectives and having an open mind about doing things the way I usually wouldn’t.

- **If you could do it all over, what would you do differently?**

I could have thought of ways to put our robot sets in front of a real human audience as often as possible, even if we weren't confident of it doing well. It was only after we got some real feedback from an audience at Maker Faire/Expo, we started to understand some of the flaws in our system. If we had started demo-ing the smallest of sets to a small audience, we could have gotten more feedback in order to iterate toward a better performing and more coherent system.

I would have also tried to know more about the research aspect of the project *right when I started working on it.* More specifically, what kind of data is useful in research, and what is not.

### 11.3 Arthur Shing

- **What technical information did you learn?**

  In terms of technical information, I learned about researching and using APIs that have little documentation, and implementing them into a project. I also learned about the research process, and how project goals, requirements, or expectations are adjusted throughout the process. Additionally, I learned about text-to-speech software limitations, and how some of these limitations can be overridden through manipulations that are provided by the software. I also learned about the limitations of audio sensors, such as their inability to detect differences due to the acoustics of their environment. I also learned how to use Choregraphe, and how to program a NAO robot through it.

- **What non-technical information did you learn?**

  Non-technical information I learned includes our expo audience analysis results. We found that people who have never seen the robot before enjoyed the robot's interactions, physical appearance, and mannerisms. We also found that people found audience interaction to be the most entertaining part of having a robot comedian, and that the context of a joke (robot/human, jobs/aging/romance) have little to do with the reception of the joke.

  In terms of comedy, I learned about routines that stand-up comedians use to up their entertainment value. I also learned about how the structure of a comedy set can affect its reception.

  In terms of Human-Robot Interaction, I learned about how quirks and mannerisms can seemingly give like to a robot's character, and make interactions more enjoyable.

- **What have you learned about project work?**

  In terms of working in projects, I learned that it is helpful to have everyone on the team on the same page at all times. I also learned about the necessity of having group members help each other outin times of need, as the project as a whole often depends on all the separate parts working together. I learned that it is also important to do your best to stick with the projected timeline and deadlines, but often project progress may fall short and goal expectations are set too high. I learned about the importance of acknowledging your own limits, with regards to time, productivity, and skill in working on large projects. I also learned about the importance of reviewing the client's requirements, to have well-attuned priorities.

- **What have you learned about project management?**

  In terms of project management, I learned that prioritizing goals, and splitting up work can make projects more productive. I learned that underestimating our team's capabilities could result in more productive uses of time, more room for ideas to foster, and a higher quality product. On the other hand, overestimating our capabilities led to our resources being stretched too thin across a multitude of aspects of the project, and made the process rushed, stressful, and not of high quality.

- **What have you learned about working in teams?**

  I have learned that it is important to listen to what your teammates are saying, and to be on the same page as everyone else on the team. I also learned that it is important to rely on aid from your teammates when there are aspects of the project that you are having difficulty with.

- **If you could do it all over, what would you do differently?**

  I would have chosen to negotiate with our client so that only one or two research areas would be our goal, as three was too much for our team. I also would have made a larger effort to grab joke ideas from friends, and would have chosen to change my research area to robot vs humanlike movements. It was much too hard of a literary problem to create variations of jokes that weren't created to have variations. I also would have spent more time developing with my teammates in their portions of the project, as it was sometimes difficult to throw our separate portions of the project together.

## 12 Glossary

Algorithm

The software program that receives input to make an optimized choice; in this project, the algorithm is in context to the adaptation program.

Animating

The NAO robot can be programmed to animate and move while speaking. This can be used to improve non-verbal communication between the robot and the audience.

API

Application Programming Interface

Branch

Looking at the graph (edges and nodes) of a performance, a branch is a decision choice made by the algorithm.

Choregraphe

Software used to program behavior and performance sets, made by SoftBank Robotics

Closing Joke

The final joke in a performance; this is helpful if it is a successful joke to end on a good note.

Crowd-work

Part of a Comedian's performance that involves content from the current audience

HRI

Human-Robot interaction

NAO

Model of Robot that will be used as the Comedian Agent, made by SoftBank Robotics

SDK

Software Development Kit

SoftBank Robotics

Manufacturer of the NAO robot, NAOqi API, and Choregraphe software

Seed Jokes

Set of three jokes that initialize the adaptation algorithm.

Set

Short for "Stand-up" set; this may also be used to describe to the collection of jokes: "A Set of Jokes"

Tree

This is the path through the set of jokes the algorithm took during a performance

## References

[1] S. Anand, J. Luetke, N. Venkatesh, and D. Wong, "Designing expressive behaviors to improve human-robot relationships," *World Academy of Science, Engineering and Technology, International Journal of Mechanical and Mechatronics Engineering*, vol. 4, no. 8, 2017.

[2] H. Knight, "Eight lessons learned about non-verbal interactions through robot theater," in *International Conference on Social Robotics*. Springer, 2011, pp. 42–51.

[3] K. Katevas, P. G. Healey, and M. T. Harris, "Robot comedy lab: experimenting with the social dynamics of live performance," *Frontiers in psychology*, vol. 6, 2015.

[4] J. Sjöbergh and K. Araki, “Robots make things funnier.” in *JSAI*. Springer, 2008, pp. 306–313.

[5] L. C. J. S. Chris Rock, Ricky Gervais, “Talking funny,” Home Box Office, dec 2012.

[6] K. Katevas, P. G. Healey, and M. T. Harris, “Robot stand-up: engineering a comic performance,” in *Proceedings of the Workshop on Humanoid Robots and Creativity at the IEEE-RAS International Conference on Humanoid Robots Humanoids (Madrid)*, 2014.

[7] A. Ernst, T. Ruf, and C. Kueblbeck, “A modular framework to detect and analyze faces for audience measurement systems,” in *2nd Workshop on Pervasive Advertising at Informatik*, 2009, pp. 75–87.

[8] G. Hoffman, “Anticipation in human-robot interaction.” in *AAAI Spring Symposium: It’s All in the Timing*, 2010.

[9] P. H. Kahn, N. G. Freier, T. Kanda, H. Ishiguro, J. H. Ruckert, R. L. Severson, and S. K. Kane, “Design patterns for sociality in human-robot interaction,” in *Proceedings of the 3rd ACM/IEEE International Conference on Human Robot Interaction*, ser. HRI ’08. New York, NY, USA: ACM, 2008, pp. 97–104. [Online]. Available: http://doi.acm.org/10.1145/1349822.1349836

[10] A. M. Leslie, “Pretense and representation: The origins of theory of mind,” *Psychological Review*, vol. 94, no. 4, pp. 412–426, 1987.

[11] B. Scassellati, “Theory of mind for a humanoid robot,” *Autonomous Robots*, vol. 12, no. 1, pp. 13–24, 2002.

[12] “Aldebaran documentation.” [Online]. Available: http://doc.aldebaran.com/2-1/naoqi/audio/index.html

[13] “Aldebaran documentation.” [Online]. Available: http://doc.aldebaran.com/2-1/naoqi/audio/alsounddetection-api.html

[14] “Aldebaran documentation.” [Online]. Available: http://doc.aldebaran.com/2-1/naoqi/audio/alsoundlocalization.html

[15] J. Palmer and J. Palmer, *Taking Humour Seriously*, ser. Taking Humour Seriously. Taylor & Francis, 1993. [Online]. Available: https://books.google.com/books?id=hEKuu7nSdDoC

[16] “Aldebaran documentation.” [Online]. Available: http://doc.aldebaran.com/2-1/dev/programming_index.html

[17] “Aldebaran documentation.” [Online]. Available: http://doc.aldebaran.com/2-1/software/monitor/index.html

[18] “Welcome to python.org.” [Online]. Available: https://www.python.org/

[19] “Aldebaran documentation.” [Online]. Available: http://doc.aldebaran.com/2-1/dev/programming_index.html

[20] “Python sdk - overview.” [Online]. Available: http://doc.aldebaran.com/2-4/dev/python/intro_python.html

[21] “Aldebaran documentation.” [Online]. Available: http://doc.aldebaran.com/2-1/dev/js/index.html

[22] “Natural language toolkit.” [Online]. Available: http://www.nltk.org/

[23] Riccardoscalco, “riccardoscalco/pykov,” Oct 2017. [Online]. Available: https://github.com/riccardoscalco/Pykov

[24] Vicapow, “Markov chains explained visually.” [Online]. Available: http://setosa.io/ev/markov-chains/

[25] “spacy - industrial-strength natural language processing in python.” [Online]. Available: https://spacy.io/

[26] “Textblob: Simplified text processing.” [Online]. Available: http://textblob.readthedocs.io/en/dev/

[27] “Socket.io.” [Online]. Available: https://socket.io/

[28] A. Asrani, “Capstone 2017,” OneNote, AKARobotics.

[29] E. McCarthy, “Craig ferguson’s new mythbuster robot sidekick,” 2010. [Online]. Available: http://www.popularmechanics.com/science/a5473/4350887/

[30] J. Welch, “Spillikin.” [Online]. Available: http://www.pipelinetheatre.com/spillikin.html

[31] “Robothespian.” [Online]. Available: https://www.engineeredarts.co.uk/robothespian/

[32] [Online]. Available: https://developer.softbankrobotics.com/us-en/showcase/pepper-takes-stage

[33] “Who is pepper?” [Online]. Available: https://www.ald.softbankrobotics.com/en/robots/pepper

[34] H. Knight, S. Satkin, V. Ramakrishna, and S. Divvala, “A savvy robot standup comic: Online learning through audience tracking,” in *Workshop paper (TEI’10)*, 2011.

[35] “Nao v5 standard edition.” [Online]. Available: https://www.robotlab.com/store/nao-standard-edition

[36] “Find out more about nao.” [Online]. Available: https://www.ald.softbankrobotics.com/en/robots/nao/find-out-more-about-nao

[37] “Sdks - nao software 1.14.5 documentation.” [Online]. Available: http://doc.aldebaran.com/1-14/dev/sdk.html

[38] “C++ sdk - nao software 1.14.5 documentation.” [Online]. Available: http://doc.aldebaran.com/1-14/dev/cpp/index.html#cpp-introduction

[39] “Python sdk - nao software 1.14.5 documentation.” [Online]. Available: http://doc.aldebaran.com/1-14/dev/python/index.html#python-introduction

[40] “Choregraphe overview - nao software 1.14.5 documentation.” [Online]. Available: http://doc.aldebaran.com/1-14/software/choregraphe/choregraphe_overview.html

[41] TH3Mjuss, “Naoqi sdk on mac,” https://github.com/TH3Mjuss/Install_pynaoqi_SDK_on_mac_10.11, 2016.

[42] mkns, “Naoqi sdk on ubuntu,” https://stackoverflow.com/questions/38705150/naoqi-library-to-python-sdk-in-ubuntu, 2017.

[43] Aldebaran, “Naoqi documentation,” http://doc.aldebaran.com/2-1/dev/python/index.html.