A Graph Based Departmental Spoken Dialogue System

By

Julia Isaac

****

Submitted in partial fulfillment

of the requirements for

Honors in the Department of Computer Science

UNION COLLEGE

March 2016

ABSTRACT

ISAAC, JULIA A Graph Based Departmental Spoken Dialogue System. Department of Computer Science, March 2016

ADVISOR: Nick Webb

Spoken dialogue systems are automatic, computer based systems that are a great way for people to receive important information. In this project, I created a spoken dialogue system that people can use to learn about the Computer Science Department at Union College. The system was built by populating an open source dialogue system using a graph based dialogue manager. I improved upon a previous working dialogue system by making the conversations sound more natural, improving the flexibility of the system and making the system more robust. To help with this process a corpus was created using about 200 different dialogues from about 20 people produced by Wizard of Oz Experiments. When the system was complete and implemented it was evaluated with 10 different participants to ensure the system is usable. The evaluation was based on the rates of task completion of people using the system, the number of turns a person has to achieve their goal and a survey given to participants. Based on the evaluation, the main issue that appears is from the speech recognition not working as well as it should. The graph based dialogue manger works well, provided the other components of the whole system works properly.

Contents

List of Figures

List of Tables

1 Introduction

The goal of this project was to complete the implementation of an existing spoken dialogue system at Union College, in order to make it a more usable system for both educational and real life scenarios. The present system was only a shell of a spoken dialogue system consisting of a GUI interface for users, speech recognition software and a Text-to-Speech engine. I implemented the dialogue manager of the system through the use of a graph. One issue with a graph based approach is its rigidness, referring to how the dialogue system understands a user's response. The system is very limited in what it can understand, especially when compared to human understanding. For example, when humans are speaking to each other, we can understand that the responses of "Yes", "Yeah" and "Sure" are all affirming responses, but this is not the case for the current system. In the current dialogue system it only understands "Yes" as an affirming response, and does not understand what "Yeah" and "Sure" mean. This means a user will not be able to move on in the conversation until they say the word "Yes", which can be extremely frustrating.

To address this issue a Wizard of Oz experiment was performed. This experiment involved users talking to what they thought was a dialogue system, which produced a variety of different dialogues that are similar to dialogues that would take place when using the final working system. The dialogues produced were then used in the final implementation. This was done by taking the dialogues and creating a corpus from it, which was used in the graph. This corpus allows the system to understand more variations of different responses the user has.

The final implementation of this spoken dialogue system contains information that can be found on the Union College Computer Science Department's website. This will allow people to have the same interactions with the dialogue system as they would with the department website. It will serve as another source of information about Union College's Computer Science Department.

The goal of this project is to have a fully functional system that people can use. To ensure the system is fully functional it was evaluated. The evaluation not only tested how well it functions, but it will also looked at the user's experience using the system. To test the functionality of the dialogue manger the rate of task completion for each user and the number of turns each user takes to complete their task was calculated. The functionality of the speech recognition was also evaluated by calculating the Word Error Rate for each

participant. To analyze the user's experience, a survey was given to participants. This survey asked about the user's overall experience, their ability to navigate the system, the answer quality, the quality of the voice used as well as any comments or suggests they have for the system.

Overall, this project produced a fully functional system with a well working dialogue manger.

1.1 Background and Related Work

A dialogue system or conversational agent (CA) is a computer system that communicates and converses with a human in a coherent way. Some examples of current dialogue systems are automated voice message systems, such as Julie from Amtrak[1] and VoiceTone®from AT&T[2] . Julie from Amtrak is a system that communicates with customers to help them find and schedule transportation for their trips. While VoiceTone®from AT&T is part of AT&T's customer service department. It is a system designed to provide customer service to AT&T customers without customers ever talking to a human agent.[2]

There are two types of dialogue systems, task-oriented and non-task-oriented. In a task-oriented system, the goal of the conversation it to accomplish a task, such as booking a hotel or planning a family trip.[3] Julie from Amtrak and VoiceTone®from AT&T are both examples of task-oriented systems. In a non-task-oriented system, there is no explicit task that needs to be solved, so the point of a non-task-oriented system is for conversational interactions, such as talking about someone's day or other forms of small talk.[3] In general, a dialogue system is a computer system that is designed to have conversations with users whether or not the goal of the conversation is to accomplish task.

There are many different parts that make up a spoken dialogue system. The structure of a dialogue system can be see in Figure 1 and involves: Speech Recognition, Language Understanding, Dialogue Manager (Dialog Control and Dialog Context Model), Response Generation, and Text to Speech Synthesis. Each of these parts plays a crucial role in the overall operations of a dialogue system. My work centers on the dialogue manager, which is the central part of any dialogue system.

An important piece of a dialogue system is the dialogue manager. The dialogue manager is the part of

[1]Julie from Amtrak can be reached at 1-800-USA-RAIL

[2]To learn more about VoiceTone®, please visit `http://www.corp.att.com/gov/solution/network_services/app_nw/contact_mgmt.html`

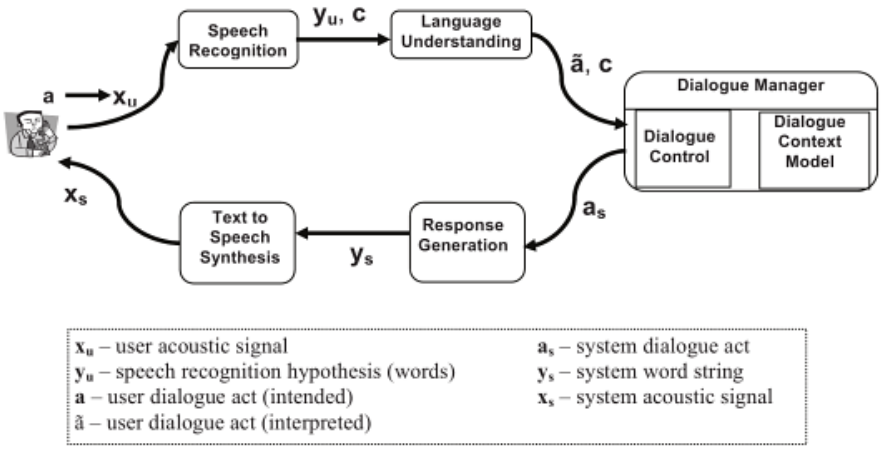

Figure 1: The structure of a Dialogue System [3]

the system that takes in the user's input and produces an appropriate response back. To do this, the manger has to first understand the data it is given and figure out what information the user is looking for. Then it has to find the correct information that it needs to give back to the user. The last step is to then generate a response that the system will say back to the user. The dialogue manager is like the brain of the dialogue system, it is the part of the system that holds information and distributes the information when needed.

The dialogue manager is comprised of the dialogue control and the dialogue context model. The dialogue control deals with the flow of control in the dialogue between the user and computer system. While the job of the dialogue context model is to mange the information that is relevant to the current conversation between the user and the system. This information is generally includes the information that has been passed between the user and system during the conversation.

One way to implement a dialogue manager is through a graph based implementation. The way this implementation works is that within the graph the action or speech that the system says is represented by the states, and the responses that the users say are the edges. By using this a graph based implementation, the transitions to each state in the graph will be caused by the responses or inputs from the user. For example, in Figure 2, at state A there two different states, B and C, the system can transition to. This transition is based on what the user's response is to what the system says at state A. This procedure will

continue until the user is done communicating with the dialogue system. A graph based implementation involves a system-directed or system initiative dialogue, which means the system will prompt the user with certain information to talk or ask about. This means that the system has all or most of the control in dictating the conversation.

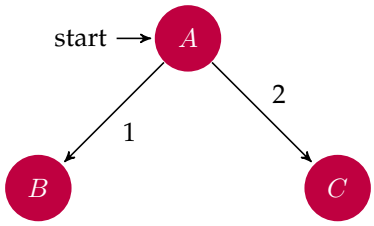

Figure 2: Example Graph

Another way to implement a dialogue manager is through a frame based implementation. A frame based implementation is made up of slots, which will be filled with inputs given by the user.[3] For example, if the purpose of the dialogue system is to find information about classes, different slots would be: course, professor and location. At first all of the slots will be empty or unknown, but if a user says: "Where is Professor Webb's CSC 103 class?", the slots course and professor will be filled with "CSC 103" and "Professor Webb", respectively. Since the location slot is still unknown the system knows that, the location of the class is the information the user is looking for.

When compared to a frame based approach the graph based implementation is very brittle and ridged because in a frame based system there is more flexibility available in the dialogue control. This is because the slots within the frame can be filled in with different combinations and in different orders from users. Graphs, on the other hand, can only handle certain information at a time. For example, if a user gives a graph based system information about a professor, but it is expecting information about a course, the system will not know how to handle the information and possibly break. The issue with the frame based system is that the algorithm to figure out the system's next response is very complex. Although, graph based systems are not as flexible as frame based systems, graph implementations are well suited for systems that have well-structured tasks, which have a set of known questions that it will be asked.[3]

The dialogue system I will be implementing will be a graph based system. This implementation was

chosen because this system is designed with well-structured tasks in mind.

Historically, there are a number of existing dialogue systems, such as GULAN, Carnegie Mellon University's Air Travel Information Service (ATIS) task and a system by Roy, Pineau and Thrun.

A group at KTH in Sweden developed an educational dialogue system, GULAN[7], which is used in language technology courses to aid students in their understanding of the different parts of a dialogue system. GULAN was designed to look up facilities in Stockholm using an online "Swedish Yellow Pages". The dialogue system was implemented using a initiative-response dialogue, which contains initiatives and responses. This creates a dialogue in a tree shaped structure, similar to a graph based implementation. The goal of this system was to help students understand the different parts that make up a dialogue system.[7]

Carnegie Mellon University has developed serval spoken language systems, which can recognize and understand spontaneous speech. One of their systems is geared toward extracting the information given by the user that is relevant to the task at hand. The researchers used a flexible frame based parser in their system, which allows for high accuracies and robustness. A flexible frame based parser is a type of implementation that uses a frame to represent the information the dialogue system holds. The frame is comprised of slots that will be filled by the information given by the user.[3] When compared to a graph based implementation, a frame based implementation is much more flexible and robust because it can handle a much larger range of inputs at any given time. Carnegie Mellon University took their system and implemented it to Air Travel Information Service (ATIS) task.[15]

Roy, Pineau and Thrun developed a graph based dialogue system using a Partially Observable Markov Decision Process (POMDP) styled implementation, instead of a Markov Decision Processes (MDPs) styled implementation. By using a POMDP-style approach they changed what the state represents. Normally, in a graph based MDP-styled implementation the states of the graph represent the system's responses to the users. In the POMDP-style, the states represent the users' responses to the system. In a MDP-style approach ambiguous speech input is difficult to handle, which can cause a lot of errors. Using the POMDP-style, Roy, Pineau and Thrun were able to reduce the number of mistakes made by the dialogue system.[12]

Other work that has been done is the creation of the Open Source Dialogue System (OSDS), which is a simple graph based dialogue system at Union College. OSDS has three major components: Google

Speech, a graph and MaryTTS/Festival. Google Speech is the system used to translate audio data into text. MaryTTS and Festival are two text-to-speech systems that OSDS can use and they are used to give users audio responses from the system. The last part of OSDS is graph, which is the dialogue manager. Its job is to decide what response should be said to the user based off the inputs given. The goal of this system was to create a dialogue system for people to interact with in a number of different ways, such as answering questions about the Computer Science Department.

2 Methods

In order to make a working graph based spoken dialogue system, the graph was first planned out. Next, a Wizard of Oz experiment was preformed to help gather the keywords needed to build the graph. Once the Wizard of Oz experiment was complete the dialogue manger was implemented. Finally, when the implementation was complete the system was evaluated to see how well it performs.

2.1 Building the Graph

To make this dialogue system, the graph must be built for the current OSDS system. The graph will contain information about different topics that are found on the Computer Science Department's Website. The different topics the system will know about are Computer Science Department's professors, courses, facilities and events.

All of the states and transitions will be implemented to create the graph. The states of the graph will be the responses the system can say and the transitions are the keywords said by the user. In order to implement all of the states and transitions we need to know the responses and keywords that we want in the system.

The responses or states in the graph are known because the information will be from the Computer Science Department's website. This information will be able to answer questions such as where things are located, what courses are being offered, who is teaching which classes, what the seminars are, etc. For example, a system response at one of the states will be: "Professor Nick Webb's office is at Steinmetz 223."

To get to this response, the user will be prompted with a greeting asking what topic they would like to learn about to which they will say "People". Next, the user will transition to the "People" state from the "Greeting" state. Then the user will specify the professor they would like to learn about and in this case it will be "Professor Nick Webb". Once the user does this, the graph will ask the user for the information they are looking for. At this point the user can ask for the office location and this question will lead them to: "Professor Nick Webb's office is at Steinmetz 223."

The keywords or transitions on the other hand are much more difficult to figure out because it is hard to predict all of the different ways users will talk to the system. To help with this problem, we will be collecting data to gather as many possible keywords a user may say. This will be done through a Wizard of Oz (WoZ) experiment, which will produce user response data. From this data, we can create a corpus of keywords that can be used when implementing the transitions.

To go along with the graph a series of dictionaries were created, which stores the information about the computer science department. The structure of the dictionaries can be seen in Figure 3. The graph call upon these dictionaries to get the information that will be given to the users. This allows the information to be easily changed if need be.

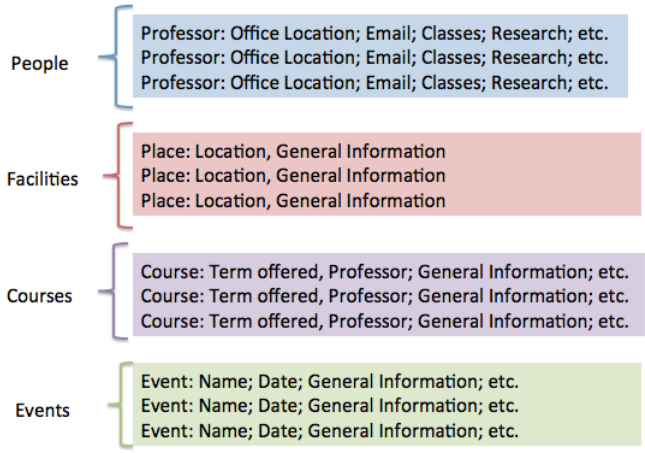

Figure 3: The Structure of the Dictionaries

2.2 Wizard of Oz

The objective of Wizard of Oz step was to create a corpus that contains responses to possible questions and statements from the dialogue system from users. The corpus was created by using the data that was produced in the WoZ experiment. The WoZ experiment is a collaboration with fellow student, Daniel Wolf. Performing a Wizard of Oz experiment allows us to collect realistic responses people will have to a dialogue system. This is important because people speak differently to a dialogue system than they do to another person.

This experiment works with two people, the user and the wizard. The user was a participant and the wizard was Daniel or myself. The user would talk to our WoZ software, and to them it just seemed like an automatic dialogue system. The wizard was in another room and they were controlling what the responses the user would hear. The wizard was acting as the dialogue manager in the WoZ dialogue system.

2.2.1 Experiment Setup

The experiment took place in the HCI Lab and HCI Control Room within the CROCHET Lab.

The user was placed in the HCI lab. In the lab, user was placed in front of a microphone, which sends the user's input into the WoZ software. The users were given various index cards with information they needed to learn about. An example index card would have: "Given Professor Mick Wubb; Find: Research". To get this information they would ask the software system through the microphone.

The Wizard was in the HCI Control Room. The wizard was placed in front of a computer with the wizard of oz software on it. The wizard also had information about a fake computer science department, which was used to answer the user's question. Figure 4 shows how the wizard and user was set up.

2.2.2 Protocol

Participants would arrive at a pre-determined time and they were given an informed-consent form before the experiment begins. Next the participants were placed into the HCI lab, where they were given instructions on how to use the software and directions on what they will be doing.

The user was given ten different scenarios or items to learn about through the "dialogue system" about

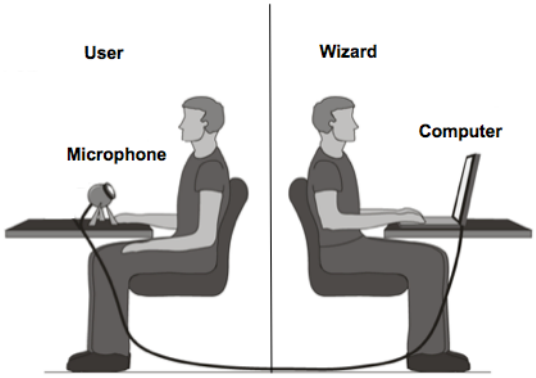

Figure 4: Basic Setup of the Wizard of Oz Experiment [1]

a fake computer science department that is similar to the department at Union College. A fake computer science department was used to ensure participants who are similar with the Computer Science Department needed to use the system to find information and could not gather information from their general knowledge of the Computer Science Department. Some scenarios the users were given are: Professor Mick Wubb, Office location and CS 109 Programming with Matrices, Professor. In these scenarios, users needed to find Professor Mick Wubb's office location and the Professor who teaches CS 109. The user would then work through the ten scenarios they are given one at time by asking a series of questions until they receive the information they are looking for. Once they got the information, the user would write down the information they learned. While the user was asking the questions to the software the wizard was in the HCI Control Room answering their questions through the software. Once all ten scenarios have been run through the participant was debriefed.

2.2.3 Software

The wizard of oz software is a fake dialogue system. The main purpose of the software is to be the intermediate step between the user and wizard, so the user and wizard do not directly interact with each other. The user would be interacting with the software, just like they would with a normal dialogue system. The software uses google speech to take in the responses of the users and then replay the responses for the wiz-

ard. The wizard would then use the information given to them about the fake department to answer the question. To answer the question, the wizard would choose an already recorded a response or answer in the software. The software would then take the wizard's response and use a Text-to-Speech (TTS) system to relay that information. The Text-to-Speech system that was used was Festival, which used the voice British Male Voice (voice_rab_diphone).

The user could only begin interacting with the system after the system produced a beep noise. This indicated that the system is ready for the user to begin interacting with the system again. To interact with the system the user would hit the enter button when they wanted to begin talking and then hit enter again to signal that they have finished talking.

The software recorded a text version of the conversation as well as audio recording of the participants.

2.2.4 Data Collection

The data that was recorded is the conversation occurring between the user and the wizard. The data was recorded through a microphone and through the software. The responses were recorded audibly and through text generated by the audio. We recorded both the user and wizard responses to get a better idea of how humans talk and interact with a dialogue system.

We collected data from the user using a microphone to get an audio recording as well as the output from the Google Speech. This was done for a few reasons. We have an audio recording of the user to get a good understanding of how humans talk and interact with a dialogue system. The audio recording helped us decide what should be used as keywords to be used to signal transitions. This was in hopes to have a smoother conversation. We collected the Google Speech outcome to help translate the audio recording into writing. The other reason we collected the Google Speech output is to help us see how accurate the Google Speech is to the actual responses. This helped us understand realistically what the inputs to the dialogue manger would be.

We collected the responses the wizard uses to answer the user's question. This data was collected to get a better idea of how a human would answer the questions the dialogue system would be answering. This was in hopes to have the conversation with the dialogue system seem more natural. We are hoping that by

having a more natural sounding responses the dialogue system will seem less robotic and stiff.

Other than collecting conversation based data, information about how users interacted with the system was also recorded. This information was important because it helped dictate any improvements that should be made in the dialogue system to make the user's experience better.

2.3 Implementation

To implement the dialogue manager a series of smaller graphs were created based off information about the Union College Computer Science Department and the data collected form the Wizard of Oz data. A script was also created to connect all of the graphs together. In addition to the dialogue manager, other implementations were added to help improve the user experience.

2.3.1 Implementing the Dialogue Manager

Five different graphs were created to help divide the different topics the system can talk about into smaller and more manageable sections. These smaller graphs can then be connected together to make the full graph that would be used has the dialogue manager. Figure 5 shows the different graphs that were created and how they connected together. The corpus of keywords created from the data produced by Wizard of Oz experiment was used to help make the transitions within each of the graphs. This was done by adding in the keywords found into the graphs as transitions, which allows users to say a larger variety of things that allows the conversation to continue. This happens because when certain a keyword is said by the user it will trigger a transition to the correct state, which allows the system to give a correct response to a user. Therefore the more keywords available the better the chances of a user saying the correct keyword to move the conversation along.

The script that was created connected the graphs together and allowed users to move from topic to topic. The graphs are connected in a similar way was the states within graphs are to each other. Keywords indicate which topic the user want to learn more about, which would trigger which graph was to be used. Another item the script controlled was any clarification issues needed by either the system or the user. This was done by sending the same message to the user if the user asked for the answer or response to be

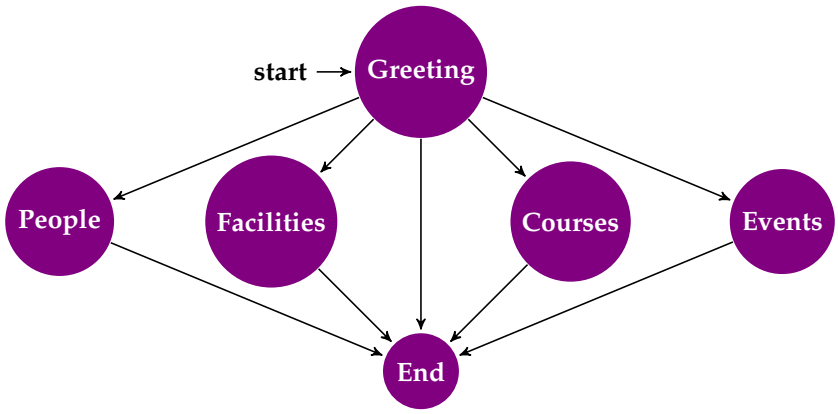

Figure 5: Basic Graph Structure with the different topics the Dialogue System can talk about

repeated. The system also resent the same message if the system did not understand the response given by the system.

2.3.2 Implementations to Help User Experience

In addition to the dialogue manager, other implementations were made to improve the user experience. The implementations made were altering how users interact with the system and the voice used by the system. These implementations were put into place due to observations taken from the Wizard of Oz experiment.

In the working dialogue system, the user interacts with the dialogue system by talking back to the system once it is done talking and then hitting the enter button when they are done talking. This was done for two reasons. The first is that before the user needed to wait for the system to make a beep noise before they could interact with the system. The issue with the beep was it would either occur in the middle or after the system's response. The issue with the beep occurring in the middle of the system's response is it can make it difficult for users to understand the system's response because it is louder than the voice from the Text-to-Speech engine. On the other hand when the beep comes after the system's response, it does not always occur directly after the system is done talking. This causes the user to begin interacting prematurely, which can then break the system. The second issue with how the user interacted with the system during the Wizard of Oz experiment is needing the user to hit the enter button to begin talking and again when they are done talking. This was an issue because users would commonly forget to hit the enter button to

before they started talking, which would cause the system to not record the user's response.

To implement this change, the beep was eliminated and the trigger to start talking was removed. The dialogue system is now implemented to start recording once the system is done recording, which is done by putting the system to sleep or pausing the system while the Text-to-Speech engine is speaking. This stops the recording from taken place until the system is finished talking. Without the pause the system will immediately start to record the user once the message to the user is sent to the Text-to-Speech software, which causes the system to record itself in addition to the user. When the system records itself saying different keywords it causes the wrong transitions to occur and gives the wrong message to the user. The recording then ends when the user hits the enter button. This change helps fix the issues previously mentioned as well as allow users to have a more natural conversation.

The other implementation that was made was a new Text-to-Speech engine added, which allows for different participants to have different voices. The ability to change voices was added to test out which Text-to-Speech engine would be the best to use. The Text-to-Speech engines were tested because during the Wizard of Oz experiment it was observed that voice produced by the Festival Text-to-Speech engine was not always easily understood. This allowed us to see which voice was more understandable by users.

2.4 Evaluation Experiment

The objective of the evaluation experiments was to test how well the system works and it make sure the system is useable. To see how well the system works, a number of evaluation measures will be calculated. The rate of number of tasks completed, the number of turns needed to achieve goal and participant surveys. These measurements show how well the system itself is working as well as show us how users feel about the system.

This experiment will consist of the user talking to the completed system using the graph built as the dialogue manger.

2.4.1 Experiment Setup

The experiment set up was very similar to the Wizard of Oz experiment setup. The evaluation experiment also took place in the HCI Room and the HCI Control Room in the CROCHET Lab.

The user was placed in the HCI lab. In the lab, user was placed in front of a microphone, which sends the user's input into the dialogue system. The users were given a sheet of paper, which contains ten different pieces of information they need to learn about. To get this information they will ask the software system through the microphone.

The HCI Control room contained the computer, which has dialogue system software. When the experiment was taking place, I was going between both the HCI Lab and the HCI Control Room. While in the HCI Lab, I would answer any questions or concerns users had while using the software. While in the HCI Control Room, I would monitor the system and make sure it was functioning properly.

2.4.2 Protocol

Similar to the Wizard of Oz protocol, participants were placed into the HCI lab and they were given an informed-consent form before the experiment begins. Next the participants were given instructions on how to use the software and directions on what they would doing.

The user was given ten different scenarios or items to learn about through the dialogue system about the Union College Computer Science Department. The scenarios was similar to those from the Wizard of Oz experiment, where the differences stem from the scenarios being based on the Union College Computer Science Department and not a fake department. Example scenarios that the users could be given are: Professor Nick Webb, email and CS 107 Creative Computing, Professor. In these scenarios, users will need to find Professor Nick Webb's email and the Professor who teaches CS 107. The user would then work through the ten scenarios they are given one at time by asking a series of questions until they receive the information they are looking for. Once they got the information, the user would write down the information they learned.

After all ten scenarios have been run through the participant was given a survey about how the dialogue system. The survey contained four questions:

- How would you rate your experience?
- How would you rate the quality of the voice?
- How would you rate the ability to navigate the system?
- How would you rate the quality of the answers you were given?

Participants answered these questions using a scale from one to five, where one is terrible, change everything and five being great, don't change a thing. In addition to the four questions there would be a comments section at the end of the survey for participants to write down what they wish to change about the system.

Once the survey is completed, the participants were able to ask any questions they had about the study or the dialogue system.

2.4.3 Software

Unlike the Wizard of Oz software, the graph based dialogue system was used. The main purpose of this experiment is to test and improve the current dialogue system. As different problems arise as different participants use the system, improvements were made, until no more significant improvements could be made. To get the information to the graph based dialogue manger google speech recognizer was used to convert audio files of the user to text that the dialogue manger can understand. The dialogue system would then produce text response based off of the user input. This text response was then sent to a Text-toSpeech software. Two different Text-to-Speech software's to be used to test out, which one works better. Half of the participants got Festival's British Male Voice (voice_rab_diphone) and the other half got MaryTTS's British Female Voice (dfki-poppy).

The software also recorded audio recording of the participants, as well as a text version of the conversations.

2.4.4 Data Collection

The data that was collected was the audio recordings of the participants, the text version of the conversations, information from the surveys and the information collected by each participant about the computer science department.

The audio recordings and the text version of the conversations was recorded using the microphone as well as the information collected from Google Speech recognition. This data allowed us to have copies of the conversion to help located and identify any issues the dialogue system might have.

The information collected from the surveys was collected from the participants at the end of the experiment. This feed back from the participants helped identify any improvements or issues the system has, which can not be seen through the recorded conversation. This data also allows us to see how people, who would use the system feel about the performance of the dialogue system.

The information collected by each participant about the computer science department came from the written answers participants are giving during the experiment. This information was used to check the quality of the answers given by the system. This helps ensure that the correct information is being given to users, as well as let us know if any information within the graph needs to be fixed.

3 Wizard of Oz Results

The results from the Wizard of Oz experiment helped to create a corpus of keywords that were used to within the transitions of the graph.

The Wizard of Oz experiment produced about 200 dialogues from about 20 different participants. To extract the key words from the data, I went through all of the dialogues and searched for all possible keywords that could be used in the graph. I recorded all of the keywords and organized them into the different sections of the graph they belong in. This processes created the corpus of transitions that was used in the graph. The corpus that was created can be seen in Table 1. This corpus really helped to improve the flexibility of the system because it allows for more variation in what users can say to trigger a transition and move the conversation along.

Professor's Name	Professor and Full Name	Professor and Last Name	Full Name
Events	Full Name	Just the Seminar Name	Seminar and Seminar Name
Courses	Full Name with Number	Just the Course Name	Teaching
Dates	Full Date	Date without the Year	
Looking for a Professor	Professor's Name	Teaches	Teaching
Looking for Courses	Courses	Classes	Just the Course Number
Looking for when a Course is offered	Courses	Offer/Offered	Term
Computer Science	Computer Science	CS	Comp Sci
Prerequisites	Prerequisites	Prereq	

Table 1: Corpus Created from Wizard of Oz Experiment

The dialogues produced by the Wizard of Oz experiment were very different and unrealistic from the dialogues that would occur on the graph based system. The dialogues produced in WoZ consisted of participants only asked or talked to the system once to get their answer. The only time they needed to talk more than once was for clarification issues from either the wizard or the user. While in a graph based system, the optimum number of times a user will need to talk to the system to achieve their gold is the depth of the graph, which will usually be more than 1. Other difference between the two is that in WoZ, users can just jump from topic to topic. Meaning after asking a question about a professor, a user can immediately ask a question about a event. In a graph based system, users need to start over or go back to the first level of the graph to ask a new question. These differences allowed for a larger variety of dialogues produced by the participants because generally graph based systems are more structured. Therefore if the WoZ experiment mirrored a graph based system more heavily, the variation and amount of keywords found would probably be lower.

4 Evaluation

The Evaluation experiment began as soon as the graph based dialogue manger was built and passed preliminary testing. 10 different participants used the dialogue system in hopes to find a total of 100 pieces of information about the computer science department. These dialogues were used to help improve the system, as well as see how well each part of the system works.

4.1 Dialogue System Evaluation Results

To evaluate how well the system is preforming the rate of completion and the number of turns a person needs to complete their task was calculated. The word error rate was also calculated to see how well Google Speech Recognition was performing.

The rate of completion was calculated by seeing how many scenarios a participant was able to get a correct answer for. The average rate of completion for the participants was 5.4 out of 10, which means there is on average only a 54% chance that a user will get a correct answer as seen in Table 2. There are two reasons why a user will not get a correct answer, the answer given to them from the system is wrong or the user got so frustrated with the system that they gave up trying to find the answer. The definition of what qualifies as a correct answer will be looked at shortly. Overall, the rate of completion average is also slightly skewed because the system was being improved upon as the experiment continued. Therefore the rate of completion for participants toward the end of the experiment is higher than the for the participants at the beginning of the experiment. This upward trend can be seen in Figure 6.

Average	5.4/10
Median	5.6/10
Minimum	2/10
Maximum	9/10

Table 2: Rate of Completion Statistics

Despite having an upward trend, there is a lot of variation in how many correct answers users get. There are two main reasons for this fluctuation. The first one is due to the experimenting of timing. As previously mention the system is put to sleep while the system speaking to ensure the system does not record itself in addition to the user. This experimentation of timing occurred because the pause initially implemented was not long enough and the system was recording itself. To fix this issue, there was a trail and error period of figuring out how long this pause should be. In the end, it was determined that the system should pause for one second per every three words the system says.

The other and much larger reason for the fluctuation was due to the Google Speech Recognizer because the system relies heavily on the information given to it by the recognizer. The more inaccurate the recog-

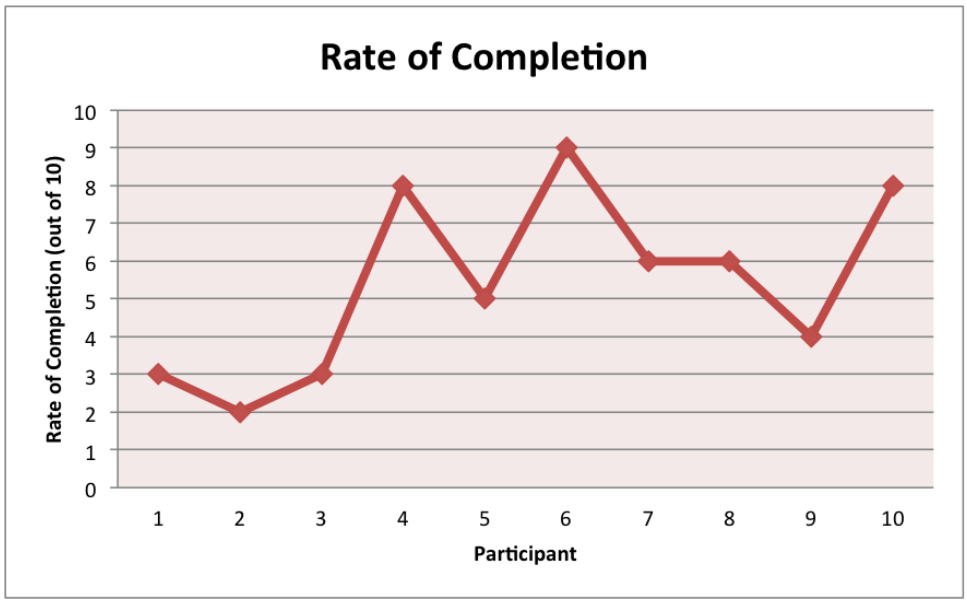

Figure 6: Rate of Completion per participant

nizer is the less likely the correct transition will be triggered in the graph. When the speech recognition is incorrect, the text received from Google Speech Recognizer does not match what the user actually says. If the text that is returned is incorrect it could mean that it does not have the keyword the user said or it could produce a keyword the user did not say. This can cause either the wrong transition or no transition to be triggered, which will cause the dialogue between the user and the system to go in the wrong direction or not progress. The more incorrect Google Speech Recognizer is the more incorrect answers will be given and the likelihood of users giving up increases.

To evaluate how well Google Speech Recognizer is working the Word Error Rate (WER) was calculated. The Word Error Rate compares a reference and a hypothesis and is calculated by finding the total number of substitutions, deletions and insertions divided by the number of words in the reference[13]. In this case the hypothesis is the text given by Google Speech Recognizer and the reference is what the user actually said. Based on participants from the evaluation the Google Speech Recognizer has an average WER of 52.7% and as can be seen in Figure 7 there is not much fluctuation in the WER value per each participant. This is important because it shows that even though different users voice may have an impact on how

well the recognizer preforms, there is not a huge impact. For the specific WER values for each user refer to Table 3. If we only look at the dialogues that produced a successful task completion, then the Word Error Rate drops to 46.3%. This shows that when the speech recognition works even slightly better, it can really make a difference because the more accurate the speech recognition is the higher the chances are of a correct keyword said by the user is passed to the system. An overall Word Error Rate of 52.7% is a significant problem when the dialogue manger relies on the speech recognizer so heavily, especially when the reported WER for Google Speech Recognition is 8%[9]. The reason for this huge difference stems from what a typically user will say while using the system. When talking to the dialogue system it is very common for a user to give one or two word answers, which is harder to convert to text because there is no context making more difficult to determine what is being said. The other issue is that many times users are using proper nouns, such as professor's names or the names of labs were are not the most commonly used words, especially when you have names like Professor Kristina Striegnitz's. Overall this high WER value becomes problematic for how well the system works.

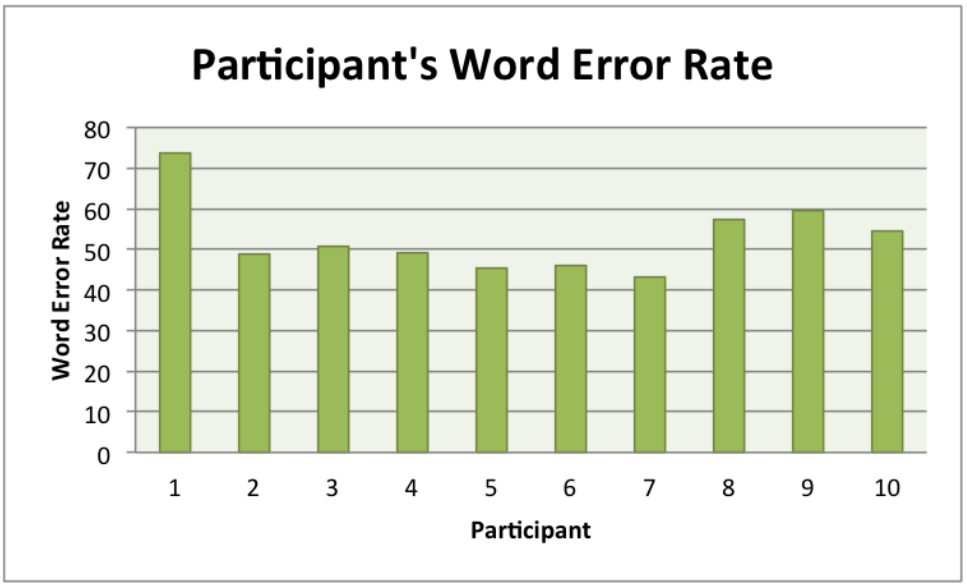

Figure 7: Average Word Error Rate of each participant

The number of turns a person has to complete their task is another indicator for how well the system is

	Average	Median
Participant 1	73.7%	81.0%
Participant 2	49.0%	50.0%
Participant 3	50.6%	52.3%
Participant 4	49.3%	0.0%
Participant 5	45.3%	42.9%
Participant 6	46.1%	50.0%
Participant 7	43.2%	38.4%
Participant 8	57.5%	66.7%
Participant 9	59.5%	66.7%
Participant 10	54.5%	61.8%
Minimum Value	43.2%	0.0%
Maximum Value	73.7%	81.0%
Overall	52.7%	52.7%
Successful Dialogues Only	46.3%	50.0%

Table 3: Word Error Rate Values

performing because it measure how many times a user needs to communicate with the system to get the information they are looking for. As one can see in Figure 8, the majority of the time the number of turns a user has is three, which is the optimum number of turns a user can have because the graph has a depth of three. On average a user will have 4.5 turns to complete their task, which can be seen in Table 4. The reason for the average been higher than three is caused by users having to occasionally repeat themselves due to the speech recognition not understanding what they said. Another reason for the inflation is from when people use the system for the first time they do not always navigate the system correctly. For example, if a user is looking for a professor's office location, they might navigate to the places or locations section of the system only to realize later that they needed to learn about people to find the information they are looking for. Also, as seen in Figure 8, the higher the number of turns the lower frequency of it occurring. This is because they more times a user needs to talk to the system, usually due to needing to repeat themselves the more frustrated they will become and more likely the user will give up using the system. In all, the fact that the average number of turns being 4.5 shows that the graph based system seems to be performing well.

Based on the Rate of Completion, Word Error Rate and Number of Turns, it is clear that the system works well, provided that the speech recognition is working properly.

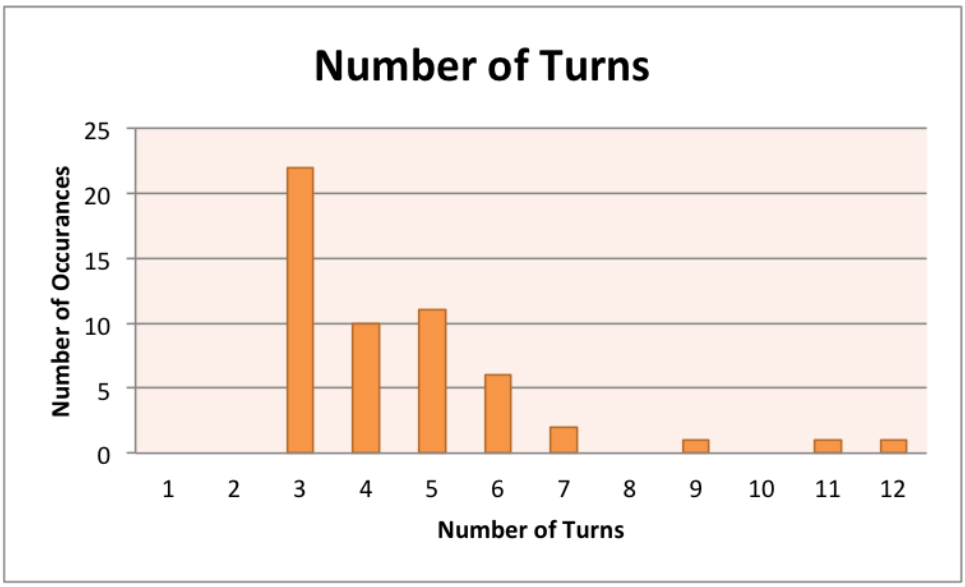

Figure 8: Occurrence of Number of Turns

Average	4.5
Median	4
Minimum	3
Maximum	12

Table 4: Number of Turns Statistics

4.2 Participant Survey Results

The surveys given to the participants at the end of the experiment were key to understanding issues the system had as well as getting a clear understanding of how users feel about using the system. It is important to get users' feed back because we need to ensure that people would actually use the system.

The first question on the survey asked how participants would rate their overall experience using the system. As seen in Table 5, the participants experience ranged the entire rating scale, where one was the low end of the scale and five being the high end of the scale. It is important to note that the low rating occurred with the beginning participants or when the system was at it's worst form. Figure 9 shows an increase in overall user experience as the participant number increases. (The participant number goes in

time order, meaning participant 1 was the first participant and participant 10 was the last participant.) This general overall experience increase is important because it shows that as the system continued to improve the user experience also improved. Despite this overall increase in rating, there is all a lot fluctuation that can be in in Figure 9. This inconsistency is caused by the same issue that affected the rate of completion, speech recognition. Since the speech recognition, does not always completely work it forces participants to consistently repeat them selfs or give them the wrong answer. This makes the participants frustrated with the system causing their overall experience to go down.

Average	2.6
Median	2.5
Minimum	1
Maximum	5

Table 5: Survey: Overall Experience Statistics

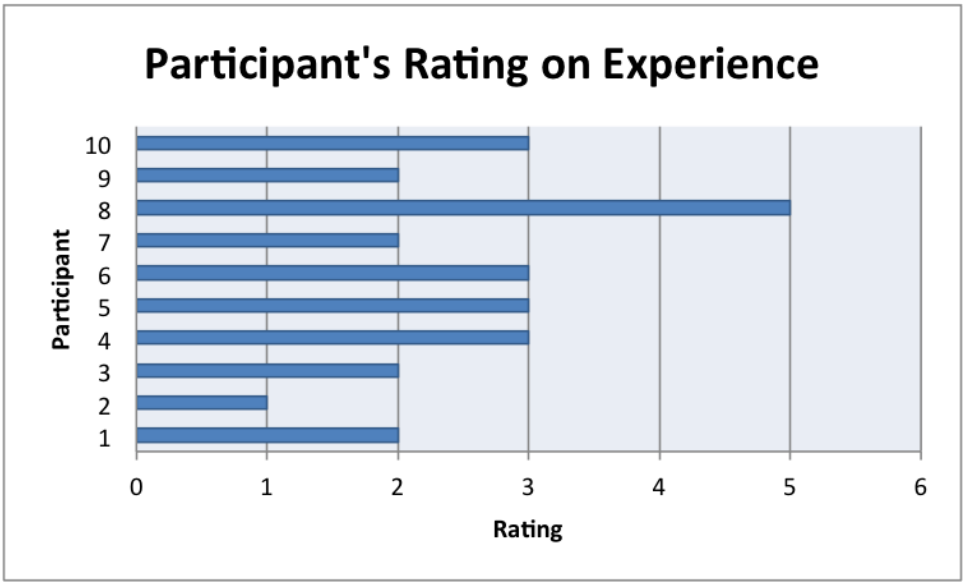

Figure 9: Overall Experience of each participant

The next question the survey asked about was how would users rate their ability to navigate the system. The average rating participants gave was a 3 out 5, which can be seen in Table 6 with other statistics about

this question. This despite this subpar rating, it is important to note that the participant's ability to navigate the system does improve as more participants test newer and more improved systems. This increase can be seen Figure 10. Also, seen in Figure 10 is a similar fluctuation in rating as seen in Figure 9. This fluctuation, like in Figure 9, is also caused by the speech recognition. When the speech recognition does not return the correct text it can cause the graph to get stuck in one state, which cause the user to be unable to move the conversation along. The other reason for this inconsistency is caused by the system recording itself and causing the wrong keywords to be returned and triggering the wrong transitions. When this happens the graph will transition to the wrong state and causes the conversation to not move in the correct direction, which decreases the ability to navigate the system.

Average	3
Median	3
Minimum	2
Maximum	4

Table 6: Survey: Ability to Navigate Statistics

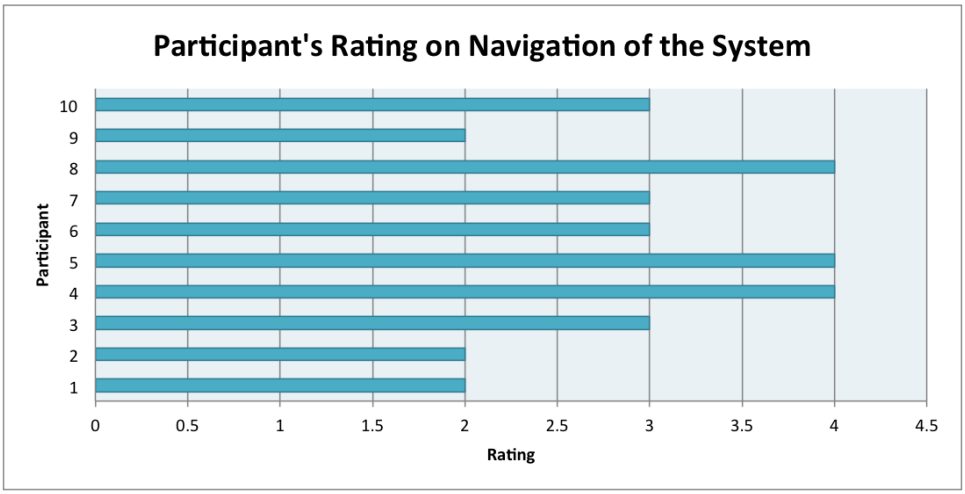

Figure 10: Ability to navigate for each participant

It is important to note that their is a correlation between participant's overall experience and their ability to navigate the system, which can be seen in Figure 11. It shows that the better a user could navigate the

system the better their experience was, which is important because it shows that improvements that were made to the system is improving the use of the system. Having an increased overall experience with the ability to navigate is also important because it shows that users are not having a bad experience no matter how well the system works.

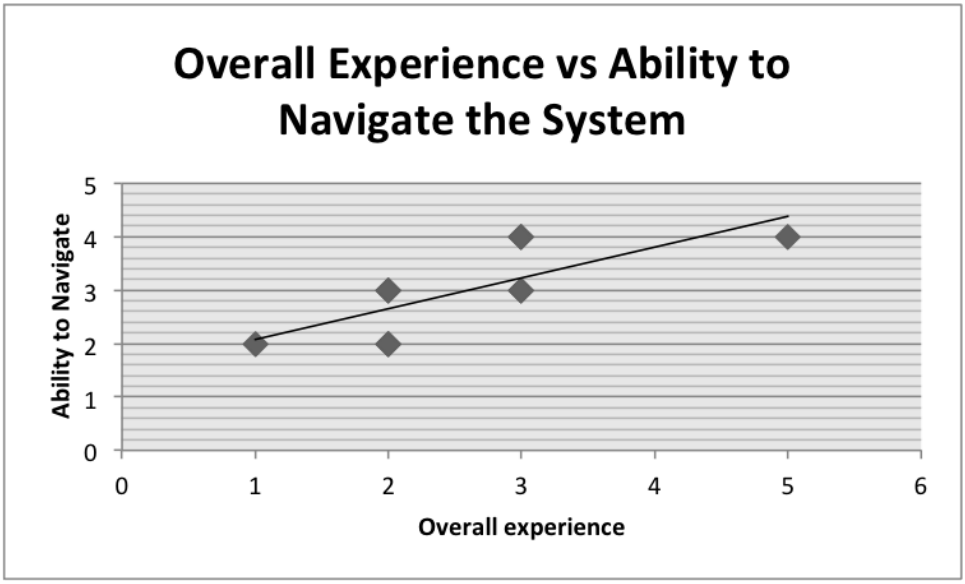

Figure 11: Overall Experience vs Ability to Navigate the System

The third question that was asked on the survey was asking participants to rate how they felt about the quality of the answers they were given. As seen in Figure 12, the participants' rating on the answer quality is very in consistent and the average rating was a 3.2, which can be seen in Table 7. This mediocre rating was a result of a two problems with the system. The first issue stems from an issue previously mentioned, speech recognition affecting the ability to navigate the system. When the system navigates incorrectly, it causes the information given to the users to be wrong. Participants can know when information given to them is wrong because they either know what the information should be or they were previously given the same information for a different topic or subject. The other reason is the quality of the voice because users can not always understand the voice used by the Text-to-Speech giving them the answers. This causes users to mishear the information giving them wrong information. It can also cause users need to ask the system

to constantly repeat itself, which many times does not help users understand the system.

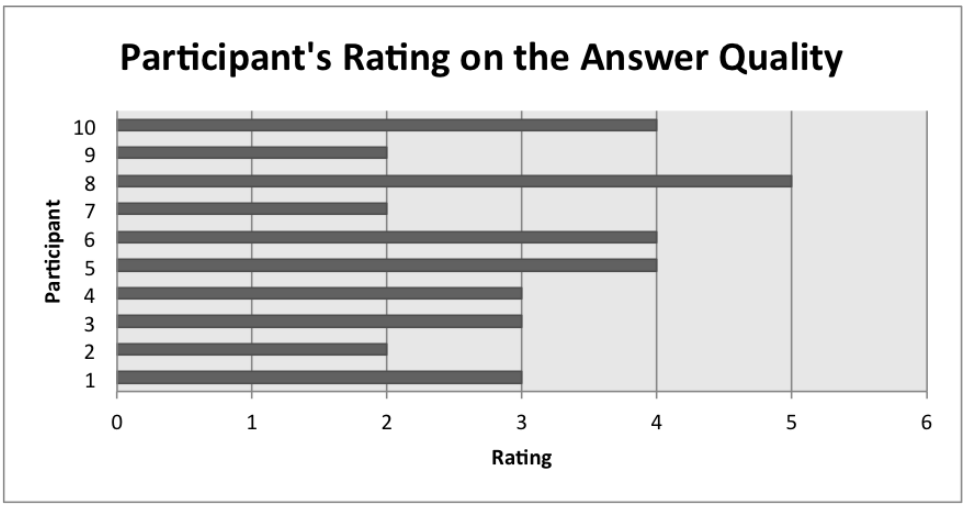

Figure 12: Answer Quality for each participant

Average	3.2
Median	3
Minimum	2
Maximum	5

Table 7: Survey: Answer Quality Statistics

The quality of the voice was the last question asked on the survey, which gave interesting insight into the issues of the voice quality. This is important because it clearly affected how people received their information and the quality of that information. During the evaluation experiment, two different voices were used from two different Text-to-Speech engines. Half of the participants used MaryTTS as their Text-to-Speech engine, which used a British Female voice, while the other half used Festival as their Text-to-Speech engine, which used a British Male voice. Using two voices were in hopes to determine which Text-to-Speech software would be better. According to the participants MaryTSS performed slightly better with an average rating of 2.9 as seen in Table 8. Festival's average rating was a 2.6. Even though the MaryTSS's rating is better, it not significantly higher. Both voices received contradicting comments. For example. one participant told that the MaryTTS voice was "cute", but a different participant thought the system using

MaryTTS “was in pain”. On the other hand, one participant said they preferred the Festival over MaryTTS, but a different participant thought that the voice produced from Festival was “scary”. Clearly, neither voice is exceptionable, but both voices are useable.

	MaryTTS	Festival
Average	2.9	2.6
Median	3	3
Minimum	2	1
Maximum	4	4

Table 8: Survey: Voice Quality Statistics

5 Conclusion

Using a graph based spoken dialogue system is a good solution to create a departmental system. The recently created system improved upon on an older system with the help of the data produced by the Wizard of Oz experiment. This data helped create a corpus of keywords, which improved the transitions used with in the graph. This helps users move more easily through the graph and decreases the amount of times users get stuck in state within the graph. As shown in the evaluation of the system, the recently build dialogue system works very well provided the speech recognition is working well and the user can understand the voice from the Text-to-Speech software. At this point in time there are no more major changes that can be made to graph or dialogue manger to improve the whole system. This project has produced a fully functioning graph based departmental dialogue system.

5.1 Future Work

In the future, some more work can be done on the graph or dialogue manger. Small improvements or additions that can be made are to connect the four different topics graphs to allow users to move from graph to graph without having to start from the beginning. Another graph improvement that can be made is to add in more transitions to different states. This will allow users to move laterally through out the graph

instead of just vertically. These improvements can help improve overall experience of the user because it will help make navigation of the system easier and more flexible.

Other work to be done, outside of improving the system, will be to combine the dialogue system with another system. This dialogue system can be combined with Anthony Pham's Virtual Agent. This would give the dialogue system a visual aspect to it as well as give the virtual agent a larger repertoire of information it can give users.

References

[1] Tina Gonsalves, Nadia Berthouze, and Matt Iacobini. The chameleon project.

[2] Narendra Gupta, Gokhan Tur, Dilek Hakkani-Tur, Srinivas Bangalore, Giuseppe Riccardi, and Mazin Gilbert. The at&t spoken language understanding system. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(1):213–222, 2006.

[3] Kristiina Jokinen and Michael McTear. Spoken dialogue systems. *Synthesis Lectures on Human Language Technologies*, 2(1):1–151, 2009.

[4] Alex Liu, Rose Sloan, Mei-Vern Then, Svetlana Stoyanchev, Julia Hirschberg, and Elizabeth Shriberg. Detecting inappropriate clarification requests in spoken dialogue systems. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 238–242, 2014.

[5] M Lynne Markus and Mark Keil. If we build it, they will come: Designing information systems that people want to use. *Sloan Management Review*, 35:11–11, 1994.

[6] Fabrizio Morbini, Eric Forbell, and Kenji Sagae. Improving classification-based natural language understanding with non-expert annotation. In *15th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 69, 2014.

[7] Anh Nguyen and Wayne Wobcke. An agent-based approach to dialogue management in personal assistants. In *Proceedings of the 10th international conference on Intelligent user interfaces*, pages 137–144. ACM, 2005.

[8] Florian Nothdurft, Felix Richter, and Wolfgang Minker. Probabilistic human-computer trust handling. In *15th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 51, 2014.

[9] Jordan Novet. Google says its speech recognition technology now has only an 8% word error rate, May 2015.

[10] Bryan L Pellom, Wayne Ward, and Sameer S Pradhan. The cu communicator: an architecture for dialogue systems. In *INTERSPEECH*, pages 723–726, 2000.

[11] Laurel D Riek. Wizard of oz studies in hri: a systematic review and new reporting guidelines. *Journal of Human-Robot Interaction*, 1(1), 2012.

[12] Nicholas Roy, Joelle Pineau, and Sebastian Thrun. Spoken dialogue management using probabilistic reasoning. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 93–100. Association for Computational Linguistics, 2000.

[13] Martin Thoma. Word error rate calculation, Nov 2013.

[14] Alexandria Katarina Vail and Kristy Elizabeth Boyer. Adapting to personality over time: Examining the effectiveness of dialogue policy progressions in task-oriented interaction. In *Proceedings of the 15th Annual SIGDIAL Meeting on Discourse and Dialogue*, pages 41–50, 2014.

[15] Wayne Ward and Sunil Issar. Recent improvements in the cmu spoken language understanding system. In *Proceedings of the workshop on Human Language Technology*, pages 213–216. Association for Computational Linguistics, 1994.