CSadvisingBot

CS 4624 Multimedia, Hypertext, and Information Access
Virginia Tech, Blacksburg VA 24061

Professor: Edward A. Fox
Client: Professor Sally Hamouda
May 7, 2022

Prepared By:
Dan Barzyk
Thomas Fitzhugh
Trey Orr
Pierre Sarabamoun
# Table of Contents

Table of Figures                                    2
Table of Tables                                    3
Abstract                                            4

## Introduction

### Requirements
- CS Advising Chatbot                                6
- Chatbot Framework                                  6
- Website Integration                                6

## Design
- System Design                                      7
- User Interface Design                              9

## Implementation
- IBM Watson                                         10
- Intents                                            10
- Dialog                                             10

## Testing                                            13

## User's Manual                                      17

## Developer's Manual                                 20
- IBM Watson                                         20
- Getting Started                                    21
- Intents                                            21
- Dialog                                             23

## Lessons Learned                                    24
- Timeline                                           24
- Problems                                           26
- Solutions                                          26
- Future Work                                        27

## Acknowledgements                                   28

## References                                         29
## Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>System Goals Diagram</td>
<td>7</td>
</tr>
<tr>
<td>Figure 2</td>
<td>User Interface Design</td>
<td>9</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Menu Navigation</td>
<td>18</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Text Navigation</td>
<td>19</td>
</tr>
<tr>
<td>Figure 5</td>
<td>IBM Watson Assistant</td>
<td>20</td>
</tr>
<tr>
<td>Figure 6</td>
<td>IBM Watson Assistant Intents</td>
<td>21</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Intent Examples</td>
<td>21</td>
</tr>
<tr>
<td>Figure 8</td>
<td>User Conversation Analytics</td>
<td>22</td>
</tr>
<tr>
<td>Figure 9</td>
<td>IBM Watson Assistant Dialog</td>
<td>23</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Original Timeline</td>
<td>25</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Updated Timeline</td>
<td>25</td>
</tr>
</tbody>
</table>
Table of Tables

Table 1  Supported Topics  .......................................................... 12
Table 2  Sample Questions and Responses  ................................. 14
Table 3  Chatbot Response Results ............................................ 15
Table 4  User Navigation Preferences ......................................... 15
Table 5  Chatbot Usability & Helpfulness .................................... 15
Table 6  Questions with Incorrect Responses  ............................. 16
Abstract

The CSadvisingBot project has a clear goal: to create a virtual assistant to answer frequently asked questions of computer science students at Virginia Tech. Advisors get asked the same questions over and over from students each semester. Most of these questions are not very complex or personalized and can be answered in a few sentences or links to more resources. These questions become a burden on advisors who have to spend their time responding to numerous emails with just a simple response. This project aims to relieve this burden from advisors by providing a service to quickly answer frequently asked questions in advising, computer science, and other frequently asked undergraduate questions. Our Client, Sally Hamouda, is an Assistant Professor in the Department of Computer Science at Virginia Tech. She proposed we create an online chatbot to help current computer science students, that can answer frequently asked questions related to advising, force add requests, funding, and more. Instead of building the chatbot from scratch, we were told to integrate a chatbot framework like LUIS or IBM Watson as a foundation of the chatbot. We decided to go with IBM Watson which uses artificial intelligence to connect user intents with dialog responses to answer a user's question as it is simple to integrate into any website, and as we hope the chatbot will be integrated into the Virginia Tech Computer Science advising website. We designed the chatbot to be primarily for current students, but it will also answer questions for prospective students, transfer students, graduate students, and parents. In order for the chatbot to be most effective we decided to implement both option navigation as well as text recognition, so the user is able to more easily get useful information that they desire.

In this report, we give a detailed description of the work completed for the project including the design, implementation, and suggestions for future work. We also provide a detailed developer’s manual and user’s manual for others to easily understand the chatbot and how to make changes as information is updated.
Introduction

As a young college student, navigating a Computer Science degree can be challenging, and finding answers to questions may be hard. Computer Science students at Virginia Tech have many questions for their advisors and professors during their time as undergraduates. There is a Virginia Tech computer science undergraduate handbook [8] which holds numerous pages of useful information for computer science students, but it is often hard to navigate and is not very well known to current students. Compiling the information from this handbook with other frequently asked questions into a chatbot will allow for students to more easily access this information by getting a quick response instead of asking an advisor. We partnered with our client, Sally Hamouda, to provide a service that computer science students can use to get answers to frequently asked questions related to advising, force add requests, funding, and more. We hope this service leads to faster and more effective advising by relieving a burden from advisors regarding frequently asked questions, leading to greater student success in Computer Science at Virginia Tech.
Requirements

CS Advising Chatbot

We were tasked with creating an advising chatbot to handle questions from newly admitted Computer Science students. This would be an online chatbot to help Computer Science students get answers for frequently asked questions related to advising, force add requests, funding, and more.

Chatbot Framework

Instead of creating a chatbot from scratch we were required to use a chatbot framework as the foundation for the chatbot. Common chatbot frameworks are LUIS [6] and IBM Watson [2]. We decided to move forward with IBM Watson because it is more of an all-in-one solution for deploying a bot to a website.

Website Integration

The chatbot must be an online chatbot that is able to be integrated with the CS Advising department website [7]. IBM Watson provides a simple script to add into the HTML of any website which will allow for simple integration and does not require any additional changes to a website.
Design

System Design

Our system was designed to satisfy the following requirements:

(1) The chatbot should provide answers to questions of prospective and current students.
(2) The chatbot should also include links to external websites that provide more in-depth information.
(3) The chatbot should provide answers to questions from parents of students or potential incoming students about the department and certain processes.
(4) The chatbot backend should allow for developers to easily add intents and modify dialog.

Figure 1. System Goals Diagram
These requirements can be broken down into goals containing tasks and subtasks (Figure 1). For any user, the chatbot must answer questions about courses or departments, give information about general tasks, and give options but also allow for text recognition. To serve current students, this means having information on checksheets, graduation requirements, or anything else found in the CS Handbook or website readily available. For parents and prospective students, this means having information about departments, advisors, transfers, and more. Finally, users and future developers should have a clear guide on how to use the software. It should be easy to update the system through a simple Intents and Dialog structure which is discussed later in the Implementation section.
User Interface Design

IBM Watson Assistant has its own user interface that is a base for each integration of an IBM Watson Assistant chatbot. As such there are limited changes we can make to the user interface. The changes we can make are:

- Assistant name
- Color (primary, secondary, accent)
- Launcher message (initial launch when a user enters the website)
- Home screen (enable or disable, custom conversation starters)
- Suggestions (enable or disable)

As this chatbot is meant for Virginia Tech students we decided to name the assistant the ‘Hokie Helper’ and use the colors of Virginia Tech: Chicago Maroon (#630031) and Burnt Orange (#CF4420). We decided to keep the home screen disabled as the home screen limits the number of conversation starters to three and we cover more than three topics in our chatbot. We also decided to keep suggestions disabled as we have no live assistant to connect to if the user needs additional help. Figure 2 shows the stages of the chatbot. First, (labeled 1) is how the chatbot looks before the user clicks on it, second (labeled 2) is how the chatbot looks if the user hasn’t clicked on the chatbot after a short period of time, and third (labeled 3) is how the chatbot looks when the user clicks on it.

Figure 2. User Interface Design
Implementation

IBM Watson

Our chatbot makes use of IBM Watson Assistant [2], which uses artificial intelligence to implement chatbots that can respond to users in an accurate and timely manner. These can easily be scaled from simple personal assistants to large-scale commercial usage. Trusted by 1000s of organizations and companies, IBM Watson Assistant tracks analytics and gives insight on how the bot can be improved which is later shown in the Intents section of the Developer’s Manual. This framework allowed us to create an intelligent virtual agent that understands user intents and responds with an answer in a structured dialog.

Intents

In our chatbot, the categorization of a user’s goal is known as their intent. Each intent can be related to a group of similar questions, such as questions about course request or force add. By giving each intent several example questions, the AI can accurately relate each user question to the correct intent. Once we know what the user wants, we can supply them with the correct information that they need.

Our chatbot supports 50+ intents. The list of supported topics is seen in Table 1. For example, a user may ask the chatbot “How do I change my major?”. The chatbot will recognize that the user’s intent is to transfer majors. This leads the bot to supply the user with a dialog of information pertaining to changing their major.

These intents are backed by examples which give the bot a rough idea of what a user with that intent might input. For our previous scenario, other examples that might exist under this intent are “Changing my major”, “How to switch to CS”, or “Applying for a new major”. It is important to have as many examples under each intent as possible; IBM Watson recommends to start with at least five examples per intent. Intents are added through the intents page by the developers by simply selecting the intent you would like to train and adding your examples in the provided table (refer to the developer’s manual for more info).

Dialog

Dialog defines how the chatbot responds to users. The goal is to give as much information as possible in as short of a response as possible. Through dialog, we can supply the user with additional information like hyperlinks. The user is able to receive different dialog depending on whether they click through the chatbot or navigate textually. We gathered information from the Virginia Tech Department of Computer
Science advising page [7] and undergraduate handbook [8] to be used as dialog responses. This includes text responses with hyperlinks to other webpages for more information as well as hyperlinks to forms.
<table>
<thead>
<tr>
<th>Advising</th>
<th>Career</th>
<th>Graduate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Academic Suspension</td>
<td>Careers</td>
<td>Five Year Bachelors/Masters Degree Program (BS/MS)</td>
</tr>
<tr>
<td>Additional Resources</td>
<td>Co-op Program</td>
<td>Graduate Application</td>
</tr>
<tr>
<td>Advising Appointments</td>
<td>Handshake</td>
<td>Interested in a Graduate Degree</td>
</tr>
<tr>
<td>DARS</td>
<td>Internship Program</td>
<td>MEng</td>
</tr>
<tr>
<td>Degree Progress</td>
<td>Courses</td>
<td>MSThesis</td>
</tr>
<tr>
<td>Extracurricular Activities</td>
<td>Academic Records</td>
<td>Research</td>
</tr>
<tr>
<td>FERPA</td>
<td>Auditing a Course</td>
<td>Independent Study</td>
</tr>
<tr>
<td>Forms</td>
<td>CRN (What is a CRN)</td>
<td>Undergraduate Research</td>
</tr>
<tr>
<td>Freshman/First-year Advisors</td>
<td>Core Curriculum</td>
<td>VTURCS</td>
</tr>
<tr>
<td>Full-time Status</td>
<td>Course Offerings</td>
<td>Major/Minor</td>
</tr>
<tr>
<td>G.I. Bill Benefits</td>
<td>Course Request</td>
<td>Applying for your Degree</td>
</tr>
<tr>
<td>Hold (Blocked, Purged, and Held Registrations)</td>
<td>Course Withdrawal Policy</td>
<td>Certification</td>
</tr>
<tr>
<td>Important Dates</td>
<td>Credit by Exam</td>
<td>Change of Major</td>
</tr>
<tr>
<td>Plan of Study Worksheets</td>
<td>Drop/Add</td>
<td>Graduating with Distinction</td>
</tr>
<tr>
<td>Policy 91</td>
<td>Force Add Requests</td>
<td>Minor in Computer Science</td>
</tr>
<tr>
<td>Probation (Academic)</td>
<td>Prerequisites</td>
<td>Minors in Other Disciplines</td>
</tr>
<tr>
<td>Schedule a Visit</td>
<td>Transfer Credit</td>
<td></td>
</tr>
<tr>
<td>Second Major Advisor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Student Success</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Upper-level Advisors</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Supported Topics
Testing

In order to have a useful chatbot that can properly respond to user intents the chatbot must recognize those intents. To ensure our chatbot is recognizing intents correctly we need to do thorough testing in both types of navigation: option-based navigation and text-based input recognition.

Users start at the Main Menu for testing the options menu navigation. This is where the user is greeted and presented with categories of topics that they can select. Each main category option should correctly route the next response to that category, which will provide another selection of topics to choose from. Some categories will have subcategories which will then take the user to another list of topics to choose from, while other main categories will just give the user a selection of topics which will then give answers to those topics. To make sure each option is tested, each final node which represents a single intent should be checked to make sure the dialog response correlates to that intent. This involves going through the option navigation for each category and its subcategories. This testing does not necessarily need to involve outside users as the options don’t change. An example is seen in Figure 3, in the User Manual section where the Courses option is clicked first, then the CRN option is clicked and a dialog response is given. This process is repeated for each category and subcategory to ensure each option gives the correct dialog response.

Users also can start at the Main Menu for text-based recognition testing. Then the user will type the question they have and the chatbot should respond. Text based recognition of intents is more complex than the option menu as the artificial intelligence has to predict which intent the user is asking about. With such a variety of ways to ask the same question the chatbot may not recognize all these variations. So extensive testing from outside users is important in building more training data for the artificial intelligence to recognize these questions in the future. Also, we can add them as examples in the intents list. Sample questions and the chatbots responses can be seen in Table 2.

We created a basic testing website through Google Sites which has a Google Forum on the left and the chatbot integration on the right. It has a few questions on what the tester is trying to get an answer for, how useful the chatbot was with option and text recognition, and any comments the tester may have. This allows testers to use the chatbot and complete the form at the same time without having to switch between tabs.
<table>
<thead>
<tr>
<th>Sample Question</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is a full time student?</td>
<td>To qualify as a full-time student, you must be enrolled in at least 12 hours. Students who fall below 12 hours could lose their financial aid, be ineligible to stay in the dorms, become disqualified under NCAA sports eligibility requirements, lose insurance benefits that are based on full-time enrollment, and jeopardize international student visa status. Dropping below full time could also change your anticipated graduation date. You should discuss your decision to drop below 12 hours with your academic advisor.</td>
</tr>
<tr>
<td>What is a CRN?</td>
<td>CRN stands for &quot;course request number.&quot; This is a five-digit number assigned to each class being offered during a given semester. The first number of the CRN designates the term. (1 or 2 = Spring, 6 = Summer, 8 or 9 = Fall, 12 = Winter).</td>
</tr>
<tr>
<td>What is Drop/Add?</td>
<td>The second phase of course registration is known as drop/add and opens about a month after course request each semester. During drop/add, a student will be able to build on the courses they received through course requests to finalize the next semester's class schedule.</td>
</tr>
<tr>
<td>What is the masters of science degree?</td>
<td>The Master of Science degree provides a solid foundation in computer science while still offering flexibility to meet the needs and interests of individual students. The MS Thesis option requires 30 credits of course work of which typically 21 credits must derive from graded courses. Students in good standing typically complete this option in two years. More information can be found here.</td>
</tr>
</tbody>
</table>

Table 2. Sample Questions and Responses
The testing from the Google Forum had 45 different user testers. From those 45 form submissions we found that the chatbot gave more correct responses when the users used the option-based navigation compared to the text-based recognition. This is seen in Table 3, as the chatbot gave a correct answer 68.89% of the time for option navigation and 53.33% of the time for text recognition.

<table>
<thead>
<tr>
<th>Correct Response to User Question</th>
<th>Yes</th>
<th>No</th>
<th>Maybe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Option Based</td>
<td>68.89%</td>
<td>15.56%</td>
<td>15.56%</td>
</tr>
<tr>
<td>Text Based</td>
<td>53.33%</td>
<td>28.89%</td>
<td>17.78%</td>
</tr>
</tbody>
</table>

Table 3. Chatbot Response Results

Although the chatbot gave the correct response more in option-based navigation, users preferred the text-based recognition more than the option-based navigation with 33.33% of users preferring option based navigation and 42.22% of users preferring text-based navigation as seen in Table 4.

<table>
<thead>
<tr>
<th>User Navigation Preference</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Option Based</td>
<td>33.33%</td>
</tr>
<tr>
<td>Text Based</td>
<td>42.22%</td>
</tr>
<tr>
<td>No Preference</td>
<td>24.44%</td>
</tr>
</tbody>
</table>

Table 4. User Navigation Preferences

In terms of usability the average rating was just above 9 out of 10 as seen in Table 5. This means almost all users were able to easily understand how to use the chatbot to get a response. However, in terms of helpfulness the average rating was just above a 7 out of 10, meaning it was helpful most of the time as seen in Table 5.

<table>
<thead>
<tr>
<th>Average User Rating 1-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chatbot Usability</td>
</tr>
<tr>
<td>Chatbot Helpfulness</td>
</tr>
</tbody>
</table>

Table 5. Chatbot Usability & Helpfulness
While looking into the questions asked by users that the chatbot was not able to answer we noticed most of them were specific questions our chatbot is not intended to answer. A list of some of those questions can be found in Table 6. Our chatbot is meant to mainly answer advising questions and currently has no ability to give a personalized response. Thus giving course suggestions, stating which courses a user needs to take in a certain term, or listing courses offered in a given semester do not make sense for our chatbot to answer. Other questions such as working on interview skills and internship experience most likely would not be asked by students to their advisors since they can easily be looked up online. As such our chatbot would not answer them correctly. However, multiple users did ask when graduation is, which is a valid question that advisors and the chatbot may be asked. We did not have an intent or dialog for when graduation is, but it would be useful to add for future students.

Considering there were many questions asked that the chatbot was not intended to answer, this may be a reason why the percentage of users getting a correct response was not very high. For the users that did ask a question from the supported topics list almost all of them got a correct answer using the option navigation and most of them got a correct answer using text recognition.

<table>
<thead>
<tr>
<th>Questions with Incorrect Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graduate course suggestions</td>
</tr>
<tr>
<td>What are the courses offered in Summer 1?</td>
</tr>
<tr>
<td>What courses will I need to take in Fall 2022?</td>
</tr>
<tr>
<td>What should I do to get internships/experience?</td>
</tr>
<tr>
<td>How can I work on my interview skills?</td>
</tr>
<tr>
<td>When is graduation?</td>
</tr>
</tbody>
</table>

Table 6. Questions with Incorrect Responses
User’s Manual

Navigation Through Menu

Navigating through the menu is shown in Figure 3. First, click on the drop-down menu of the Main Menu to see all the categories supported (Steps 1 & 2). After selecting the category, you will then either be prompted with another drop-down menu or with a selection of options with new categories (Step 3). Once you have selected one specific enough, the necessary information will be displayed. Select the “Main Menu” option to be directed back to the main menu (Step 4).
Figure 3. Menu Navigation
Navigation Through Text

Navigation through text is shown in Figure 4. At any point, either at the beginning of launching the chatbot or while in the middle of menu navigation, you can enter a text inquiry at the bottom of the chatbot (Step 1) and it will attempt to locate the information you desire (Step 2). If you type something the chatbot does not understand it will communicate that and ask you to rephrase and prompt you with the main menu options again (Step 3).

Figure 4. Text Navigation
IBM Watson Assistant is an extremely useful tool for designing chatbots because of its ease of implementation and automatic UI generation for integration on custom websites. From a developer standpoint we found it very easy to learn and use, so it should not be a problem to implement future iterations. We have included documentation on how to navigate IBM Watson Assistant below. The home page after signing in is known as the Assistant page which shows all assistants that you are working on. From the Assistant page, select your desired Assistant and you are taken to that assistant's home page shown in Figure 5. Then, you may add Intents and Dialog to desired skills. In our project, we added intents and dialog to the “Advising Skills” skill.
Getting Started

Download the “Advising-Skills-dialog.json” file that should be provided to you by the client onto your desktop. Create an IBM account (IBMID) adding all the necessary information. Login to IBM cloud so that you are at the dashboard. Click on “Create resource” at the top right of your screen then select “Watson Assistant”. After creating your Watson Assistant instance click on “Create skill”, then “Dialog skill”, then finally “upload skill” and drag and drop the JSON file you previously downloaded.

To give all groupmates access to the same Watson Assistance instance navigate back to the IBM Cloud dashboard then click the “Manage” drop down menu at the top of your screen. Click on the “Access (IAM)” option. Click on the “Create access group” button at the top right of your screen. The name is whatever you want. Navigate to the “Access policies” tab and then click on “Assign Access”. Have “IAM Services selected and search for “Watson Assistant” in the dropdown menu. Add that and go back to the group. This time navigate to the “Users” section and invite all your group members.

You should be all set and ready to make changes. For any additional questions check out the IBM Watson Community Forum [3] which has user guides and more.

Intents

You can find the Intents section on the “Advising Skills” dialog page. The “Advising Skills” dialog page has navigation on the left side which includes Intents,
Entities, Dialog, Options, Analytics, Versions, and Content Catalog as shown in Figure 6. The Intents page (Figure 6) lists all intents, and when clicking on a certain intent you are shown the examples of that intent; see Figure 7. There you can create new intents and update or remove existing intents. When creating or editing a specific intent, you can add User Examples to help train the bot. The bot will take the context from any examples and direct users to the desired intent based on their input.

Also, IBM Watson Assistant tracks all conversations and stores them so you can see exactly how the chatbot responded, to make any changes if it misidentified an intent. This is found under the Analytics tab which has a User Conversations subcategory seen in Figure 8. As the chatbot correctly identifies new questions it automatically adds them as user examples to use in the future, so the chatbot will continue getting better as time goes on. As seen in Figure 8 the user typed “Credit by Exam” in which the chatbot recognized the intent as #CO-Credit-by-exam which is the related intent for credit by exam. Since it correctly identified the intent, it would add the phrase “Credit by Exam” to the user examples if it is not already in the example list.

Figure 8. User Conversation Analytics
Dialog

You can find the Dialog section on the “Advising Skills” dialog page, below the Intents section. There you can add nodes to the Assistant’s dialog tree. The chatbot will respond to the user based on the intent it perceives. You can choose different response modes, such as text, audio, video, and more. The node tree, with a selected node showing the recognition and response, is illustrated in Figure 9.
Lessons Learned

Timeline

A brief timeline is shown in Figure 10 and Figure 11. Following are details:

- Become Used to IBM Watson Assistant Interface & Set up plan moving forward - (02/17/2022)
  - Get the team on the same Watson Assistant instance and use the interface.
  - Have a clear plan on necessary steps to create a working chatbot

- Working Basic Chatbot - (03/24/2022)
  - Populate Basic Chatbot with Intents and Dialog from multiple sources (Advisors, FAQs, Student Handbook) - (02/24/2022)
  - Get an interface working with decent design to set up for user testing
  - Create user guide through documentation - (03/03/2022)
  - Test the chatbot and make sure listed intents and dialog work correctly with interface, and commence user testing
  - Take user feedback and do necessary changes (if possible) - (03/15/2022)
  - Expand Chatbot Capabilities and Intents to account for Graduate Students (BS/MS)

- Working Advanced Chatbot - (04/21/2022)
  - Expand List of Intents and Dialog
  - Look Into VT Data integration
  - Expand Chatbot Capabilities and Intents to account for Graduate Students (BS/MS)
  - Further improve UI as specified by Client (give user selectable options in a neat manner) - (04/07/2022)
  - Write up documentation for new UI for users, developers, and advisors - (04/14/2022)
  - If VT Data integration is possible duplicate current chatbot and update dialog and add intents with this new capability - (04/21/2022)
  - Conduct another round of user testing
  - Take user feedback and do necessary changes (if possible)

- Polish the Chatbot - (05/05/2022)
  - Clean up UI and make changes given by client
  - Add any other intents given by advisors or client
  - Make sure documentation is polished
  - Final round of user testing
  - Take user feedback and do necessary changes (if possible)
In early May we had planned for data integration and personalized responses based on user data as seen in Figure 10. But we realized integrating a database for the chatbot to use would take longer than the time we had remaining. So, we decided to abandon that idea and switch to a documentation approach in order for a future team to pick up the project from our progress and expand it for data integration and personalized responses as seen in Figure 11.

Figure 10. Original Timeline

Figure 11. Updated Timeline
Problems

We are currently undergoing testing to further train the bot. So far, we have gotten 45 responses from the survey on our chatbot’s site. Among those 45 users, the topics were spread out fairly evenly among Advising, Career, Courses, Graduate, and Major/Minor. For the Option Based method, 68.89% of users said the chatbot gave useful information based on their question. For the Text Based method, 53.33% of users said the chatbot gave useful information based on their question. Based on these responses, we conclude that we do not have enough examples to provide correct responses at a highly consistent level. The chatbot will misidentify a user’s intent if there are not enough predefined examples in the system corresponding to that intent.

This is mainly due to the wide variety of intents that users may have. IBM Watson recommends supplying five examples for each intent that must be identified. It was difficult to come up with a list of over 400 examples in total for 60+ intents, mainly because it requires some creativity when brainstorming different ways to say the same thing. There is an average of 7.5 examples per intent, with a low of 5 examples and a high of 17 examples. The variance is relatively high as some topics have a lot of ways a user can ask about them, whereas some topics have a limited number of ways a user can ask about them. Because the context of a question varies depending on the person asking the question, we realize our amount and breadth of our examples may not be enough to ensure an accurate categorization model.

IBM Watson takes in the provided examples for each intent (the more the better) and trains a model to recognize both keywords and phrases in the examples to be better able to identify the intents.

Our intents cover Advising, Courses, Career, Graduate, Major/Minor, and Research tops. The main area that we feel is not well covered enough is the Courses area since the bot cannot make suggestions about which courses to take but only how to Force Add a class, or drop a class, and other things linked with course management and FAQs.

Solutions

All information gathered and loaded into the chatbot was taken from three sources. We reached out to advisors with what questions they were frequently asked. We went through the CS advising FAQ page [7] and took all the questions and answers out of that. Finally, we went through the CS Handbook [8] and extracted all the questions and answers we could from that as well.

The issues listed in the “Problem” section can be solved by collecting more testing data and feedback. Each intent correctly identified during testing can be used to further train the bot in IBM Watson. We collected user data as the semester progressed
and made the necessary changes. Dr. Fox agreed to create an extra credit assignment for the class which led students to test the chatbot and leave feedback. Having more examples tied to each intent improved the bots ability to interpret a user’s needs.

Advisors could also help with testing since they deal first hand with students that need advising. They would be able to provide a lot of important feedback regarding the chatbot itself and examples to be added for certain intents. They also have the ability to go in themselves and add the examples for certain intents given that the developers give them access (view the Developer Manual for instructions).

Future Work

In the future, we would like to have the chatbot to be database-driven [5] and integrated into the Virginia Tech Computer Science Advising website, but the project has not been approved by the administration. Integration into the advising website is as simple as copying and pasting the provided embed script, which is found in the integration page of the IBM Watson Assistant, into the HTML of the advising website. The script is only a few lines of code and requires no additional changes to the website. As of now, you must travel to an external Google Site [1] we created that displays the bot interface. The instance of the chatbot running right now is a free service that allows for up to 1,000 monthly users. There is a paid plan called “Plus” [4] starting at 140 US dollars per month that allows for 1,000 monthly users and charges 14 US dollars for every additional 100 monthly users. The paid plan does ensure a 99.5% uptime. The Plus plan offers many features such as multiple language support and intent recommendations that may be useful for the Computer Science advising team. Once a chatbot is created that Virginia Tech deems completed and ready to publish, they simply just need to switch from the free service to the “Plus” service and add the code given by IBM Watson Assistant straight into the webpage’s HTML.
Acknowledgements

Dr. Edward Fox
Virginia Tech, Professor of Computer Science
fox@vt.edu

Dr. Sally Hamouda
Virginia Tech, Assistant Professor of Computer Science
shamouda@vt.edu
References


