1. Functionality of the chatbot - Description

The design of the chatbot helps alleviate student traffic sent to career advisors at the University by directing students to a bot. The bot helps students by providing career advice in two forms: either by displaying a list and description of 5 typical jobs or allowing the student to take a quiz. The quiz will ask a few basic questions then suggest one of the careers mentioned above. Users can select any of the five career options, and a brief description will appear along with a link to read more information. In terms of function, the chatbot will direct users towards a career. If a user gets stuck, they can type "help" and an appointment link is generated. Basic descriptions provide an overview of each career option, and associated links provide more information for each career suggestion.

1. Summary of outside sources

Please refer to Sources for complete citations of the reference material in this section.

The 2016 paper, The Development of a Chatterbot for Environmental Education, provided an excellent overview of how recent research handles bot interaction. The examples provided in the "Tests" section helped me adjust my bot to direct the conversation towards its goal.

2018 article Chatbots Demonstrate Diverse Range of Uses, helped me identify what made chatbots successful in the education field. It states that the main advantage of bots in this area is the bot's ability to provide follow questions and create a personalized experience. This is what helped me come up with the idea of offering a quiz for the students.

The 2018 article by University Business helped me identify that students would need direction to guarantee results. After reading about how Georgia State University issued reminders, I got the idea to have the bought schedule appointments for students. I expanded on this idea by including the "help" feature. If students were unable to get the information needed by speaking with the career services bot, then the bot would tell them to talk with an advisor.

Another 2018 article, Chatbots for Education, expresses the importance of creating chatbots to education users. The report gave me the idea to include a brief description and a link for more information. I thought about how I interact with my google home daily and realized the value it provided me through interaction with it. This robust design feature is what makes the bot useful for students.

Lastly, the 2016 report, Talking to Bots: Symbiotic Agency and the Case of Tay, opened my eyes to the complexity of open-ended communication. This gave me the insight to not force a conversation where the user asked questions. Instead, I focused on directing the conversation towards helping students identify popular computer science jobs.

1. Identified computing jobs

The bot suggests five different career paths within the field of computer science. They are as follows: software developer, cybersecurity analyst, UX designer, database administrator, and web developer. These career options can be further investigated by looking at their associated AIML files. The File names outlined in section F correlate to each suggested job. The bot has two different ways of providing students with suggestions for which field could best suit them. The first is displayed as a list where a user can click on any available career path, and they are provided with a description and link to additional information. The second way is through a series of questions. The questions attempt to determine what the student's interests through yes and no statements. At the end of 3-4 questions, a student is presented with a suggestion. Included with the advice is a brief description and link to additional reading.

1. Training Cases

Training occurred by issuing an initial deployment for testing. Users began by testing a black-box version of the bot, and the conversions are recorded. Keywords are selected from test results and are hardcoded into the bot using AIML. After reviewing common words, it became apparent that the bot needed to focus on directing the conversation instead of answering user questions. This is because the scope of user questions is vast, and answers are mostly dependent on a case by case basis.

As a result, we focused on training the bot to direct users either to begin the career advice process or to navigate to the help option. We tried to identify similarities and differences between the five available career suggestions and train the bot to go through a binary decision tree. This binary tree was traversed by asking yes or no questions. Each answer eliminates other possible outcomes until the bot has narrowed in on the user's interests. We also involved extreme cases for students who did not fit any distinct suggestions. For these cases, we recommended that they speak directly with an adviser. Below we provide two summarized examples of how the bot can identify a recommendation based on user preferences:

John is a student who enjoys programming and prefers to work in a development environment. Additionally, he also enjoys engineering, designing, installing, testing, and maintaining software systems. After learning about these traits, our bot will recommend that John investigates careers as a software developer. Another path John could take is as follows:

John is a student who enjoys design but struggles with programming. He has a fair bit of experience working with web applications and various APIs. However, John does not enjoy working with only web applications and has a vast portfolio of experience working in different project development teams. Our bot does not need to know all the specifics but can ask questions to determine most of the information stated above. As a result, the bot suggests that John investigates a career as a UX Designer.

1. AI Optimization Methods

Optimizations and improvements for the bot are realized as testing and design were applied to the initial production bot. One such realization was our decision to push away from having the bot answer questions and focus on directing the conversation. In terms of user experience, we quickly realized that to address the conversation, we must make it easy for users to respond continually. This is accomplished by implementing a button interface. This optimization feature allowed users to stay engaged and while also have a sense of control over the interaction process. Another improvement we made to the bot was its ability to reason through user answers based on the current question.

1. Installation Manual

Below we provide instructions on how to access and run the bot through the Pandorabot website.

1. On a desktop computer, navigate to [www.pandorabots.com](http://www.pandorabots.com) in your preferred web-browser.
2. Proceed to log in and go to your dashboard. On the left-hand side, find the header, "My Bots," click the plus icon on the right-hand side to create a new bot.
3. Give your bot a desired name and ensure "Blank Bot" is selected under context.
4. Select the bot name on the left-hand column and click on the Edit dropdown to navigate to Code Editor.
5. Locate the file dropdown in the code editor and select upload.
6. Proceed to upload the downloaded files that were in the zipped file named wgubot. The filenames are as follows: begin.aiml, cybersecurityanalyst.aiml, databaseadministrator.aiml, softwaredeveloper.aiml, udc.aiml, uxdesign.aiml, webdeveloper.aiml, and help.aiml.
7. From here, you can select the orange icon on the bottom right-hand side of the screen and begin interacting with the bot.
8. Effectiveness of the Bot

The bot has continued to become significantly better at directing users to possible career paths. It can effectively navigate someone to a career path and schedule them for an appointment. It does have a failsafe in place if the user is unable to reach a recommendation. This failsafe suggests that the user plans an appointment and contacts their advisor with their desired appointment time. Another example of the bot's effectiveness is its ability to provide external resources outside of the bot environment.

In terms of maintenance, the chat logs must be monitored for persistent questions. Over time, the bot should be adapted to provide direct responses for these questions instead of asking the user to schedule an appointment. This will help continually improve the function of the bot by reducing the number of meetings that are scheduled. Another critical aspect of maintenance is to monitor the usage of its suggestions. Perhaps implementing a user survey will help us determine if users find the recommendations relevant to their interests. Yet another essential aspect of the maintenance and upkeep of the bot is to ensure that it does not break or remain unresponsive for long periods. Great care must be taken to ensure that the bot is always available for it to be considered a useful tool.

1. Challenges during Development

We faced many challenges initially with building the bot. At first, we wanted to create an extensive list of questions and answers so that the bot could cover as much as possible. When soon realized that this approach was incredibly difficult to track and would lead to students becoming confused. Once this was realized, we had to restructure the bot to reduce the amount of input. This was accomplished by responding to any question with a prompt on how to navigate the chatbot. Another challenge was creating questions that could help identify students' interests. We first had to identify popular jobs that were not overlapping. Next, we had to focus on determining the similarities and differences of each field. Once this was established, we had to link related jobs and create a question tree. This binary tree could be easily traversed based on user input and was then programmed in AIML.

1. The Bot Development Environment

One strength of the bot was how easy it was to get started with building the bot. Learning AIML and how to navigate through the environment was quick and easy to understand. The vast documentation and learning resources further supplemented this. However, as you began writing the code, you realized the limits of the development environment. There were a lot of features that felt missing from the GUI. Additionally, the environment also failed to provide enough depth for their existing features, such as the media-rich tools. Overall the program was a great learning experience. However, it felt lacking in modern technology that I learned about in my studies and reading.

Sources

1. de Oliveira VT vinicius. tdo@hotmail. co., da Silva EG egilberto@uol. com. b., Silva PP patrickpsilva@gmail. c. The Development of a Chatterbot for Environmental Education. ICICTE Proceedings. July 2016:408-416. https://search.ebscohost.com/login.aspx?direct=true&db=eue&AN=119264921&site=eds-live&scope=site. Accessed January 24, 2020.
2. Mathieson SA. Chatbots Demonstrate Diverse Range of Uses. Computer Weekly. June 2018:21-25. https://search.ebscohost.com/login.aspx?direct=true&db=ofm&AN=130143340&site=eds-live&scope=site. Accessed January 24, 2020.
3. Bendici R. Rise of the machines: Artificial intelligence--led by text-based chatbots--has infiltrated campus life, helping institutions improve communication, compliance and retention. University Business. 2018;21(10):53-55. https://search.ebscohost.com/login.aspx?direct=true&db=eue&AN=132179007&site=eds-live&scope=site. Accessed January 24, 2020.
4. GEORGESCU A-A alinandrei2007@yahoo. co. Chatbots for Education - Trends, Benefits and Challenges. eLearning & Software for Education. 2018;2:195-200. doi:10.12753/2066-026X-18-097.
5. NEFF G gina. neff@oii. ox. ac. u., NAGY P peter. nagy84@gmail. co. Talking to Bots: Symbiotic Agency and the Case of Tay. International Journal of Communication (19328036). 2016;10:4915-4931. https://search.ebscohost.com/login.aspx?direct=true&db=ofm&AN=127361911&site=eds-live&scope=site. Accessed January 24, 2020.