Course Helper
[Shivam Bang, Rajat Bishnoi, Rodrigo Calvo, Praveena Hariharan] - CIS6930
Project Proposal to CIS 4930/6930 – Spoken Dialogue Systems
Fall 2021

1. Introduction

1.1 Motivation

Everything in today's world is fast paced. People want instant access to information. Students, who are getting more and more accustomed to the latest technologies, desire quick and easy solutions. To keep up with this generation, educational institutions need to adapt and build systems that can help students more effectively. Repetitive and time-consuming tasks such as inquiries about the admission process or courses offered can be automated.

A smart solution to these problems would be a dialogue system for students where they can interact with the system and find courses and faculty related information at ease. Having a spoken dialogue system as opposed to the conventional searching through the portal would actively engage the students in exploring a wide range of courses offered. This would also highly benefit new and prospective students who are unfamiliar with the institution's web portals.

1.2 Description of System

The system will be able to help users decide what courses to take in a semester. The system will be able to look up different courses offered by the CISE department at UF, tell the location, description, books required, and timing among other things. It will then also be able to search for the required books for the course in the UF library and tell the user about availability. This will make the process of searching for class related information easier. The user wouldn't have to visit the university or other websites in search of different types of information. The system would assume the persona of a librarian/administrative assistant. Vicky, a 40-year-old librarian at a local college. Graduated from a state university without debt. Hobbies include reading, watching plays and going to the beach. Cheerful and happy to help students. Often busy and values quick responses. Currently healthy and plans to retire early by 52.

1.3 Example Dialogue

User: What are the courses available this semester?
Agent: *Query Course Database* Here are some of the courses *Provides a list of courses*.
User: What is COT5405 about?
Agent: COT5405 Analysis of Algorithms is Introduction and illustration of basic techniques for designing efficient algorithms and analyzing algorithm complexity. Would you like to see the course syllabus?
User: No. Who will be teaching the subject?
Agent: *Professor*. Would you like to visit their website?
User: Sure.
Agent: *Query Professor Database and redirect to website*
User: What are the class timings?
Agent: The classes will be in *Room* on *days* at *time*.
User: What are the required textbooks for this course?
Agent: Here are the required textbooks *List of required texts*.
User: Is the textbook available in the library?
Agent: Let me check. *Query UF library to check for availability*

2. Development Plan and Goals

Project development will proceed within three phases of goals. The first set of goals, the minimal goals, are essential to have a functioning system/result. These goals will be accomplished **before** the first prototype test. The second set of goals, basic goals, are necessary to fulfill the group's vision for the project and will be accomplished before the final round robin evaluation class day. The final set of goals, stretch goals, will be accomplished if the basic goals are completed early or if the team wishes to continue system development after the round robin evaluation class day. The stretch goals are “above and beyond.”

2.1 Minimal Goals - Target October 25

- We intend to build the system using Python.
- Speech Recognition/Speech Synthesis
- Intent Classification
- The system should be able to complete some basic tasks-
    - What courses are available?
    - What is *course* about?
- Testing. Test both the speech recognition and synthesis and make sure they are recognizing and classifying intent and speech appropriately. Basic testing to make sure functionality of our system works well for happy path scenarios.

2.2 Basic Goals - Due November 8 for initial system prototype round robin (then your development continues to refine the system until final evaluation round robin Dec 6.)

- The system should be able to answer particular course information
- Course syllabus and professor details
- Filter courses (by timing or instructor) according to user input
- Information about course textbooks- are they available in the university library.
- Identify context.
- Testing. An in-depth testing of all aspects of the system. Speech and intent recognition as well speech synthesis will be tested in different environments to make sure that they work in both noisy and quiet settings. Also, we will test the system with people with different accents to make sure that the system can be used across a wide variety of users. We will also test abrupt exiting of conversation, test interruptions and irrelevant responses to make sure the system handles them seamlessly.

2.3 Stretch Goals

- Link the system with ONE.UF to add additional functionality like showing only those courses that have not already been taken by the user.
- Link with Canvas. Users can check assignments and due dates. E.g.: Check for any pending assignments due that week or send email to peers.
- Add information for other departments
- Add other students as friends to see what courses they are planning to take.
- Extend the usage beyond courses to help gather information about all Gator events and facilities.

3. Collaboration Plan

- **Meetings.** We will meet weekly on Mondays after class (2-3PM).
- **Time Allocation.** Work will be allocated task wise instead of time wise. As long as we are making steady progress in the project each individual member will decide how much time to put in to at least complete the assigned tasks for the week.
- **Source Code Management.** Git will be used for source code management. A repository has been created - Spoken Dialogue System.
- **Task Management.** We will use GitHub for managing tasks that each member is responsible for. GitHub repositories have a section that contains boards for to do, in progress and done, where each team member will add what they are working on and pick up tasks.