CS1980 – Capstone Team Project
Department of Computer Science
Spring 2021
School of Computing and Information, University of Pittsburgh

Team Project Proposal

Title

Virtual Kathy: A conversational agent for the CS Department

Mentor

Dr. Alexandros Labrinidis (labrinid@cs.pitt.edu)

Team Members

Liu, George	*georgeliu@pitt.edu*
Lin, Sophia	*sjl81@pitt.edu*
Yu, Feiya	*fey20@pitt.edu*
Liu, Pan	*pal81@pitt.edu*

1. Introduction

Background

Chatbots or conversational assistants are computer programs that are capable of interacting with the user using language-based interfaces [1]. The main objective of a Chatbot is to simulate an intelligent human conversation so that the interlocutor has an experience as closely as possible to the conversation with another real person.

There are two categories of chatbots. One category is command-based chatbots where chatbots rely on a database of replies and heuristics. The user must be very specific while asking the questions so that the bot can answer. Hence, these bots can answer a limited set of questions and cannot perform functions outside of the code.

An alternative is conversational bots based on decision trees or driven by artificial intelligence. The interface used by Chatbots is based on the structure of human conversation that is obtained through natural language processing (NLP). The NLP allows algorithms to understand, interpret and manipulate human language [2]. In addition, more advanced Chatbots are able to learn from conversations by implementing Machine or Deep Learning [3].

Motivation

AI-powered chatbots are motivated by the need of traditional websites to provide a chat facility where a bot is required to be able to chat with users and solve queries. If any school or business company is receiving lots of queries, having a chatbot on a website takes off the load from the support team. Having a chatbot clearly improves the response rate compared to the human support team. Furthermore, a chatbot can automate repetitive tasks. There can be some scenarios where a business or school receives the same queries in a day many times and the support team must respond to each query repetitively. Last but not least, one of the most important advantages of having a chatbot is that it is available at any time and every day. No matter what time it is, a user can get a query solved.

2. Project Objectives

This project is focusing on creating a conversational agent (chatbot), *Virtual Kathy*, to be used by students to get their queries responded easily from the CS department front office. The project entails:

- creating training questions (by crowdsourcing them)
- creating matching answers (some static / some through a database)
- setting up the conversational agent
- supporting “context” (i.e., follow-up questions)
- routing non-answerable questions to humans
- testing the chatbot
- submitting the new questions and answers to step one to improve the system

3. Project Planning

Programming Language

In this project, we expect to use Python as our primary programming language, along with TensorFlow (a Deep Learning library) and NLTK (a Natural Language Processing toolkit), and some helpful libraries. Additionally, PostgreSQL will be the database management system to store necessary data.

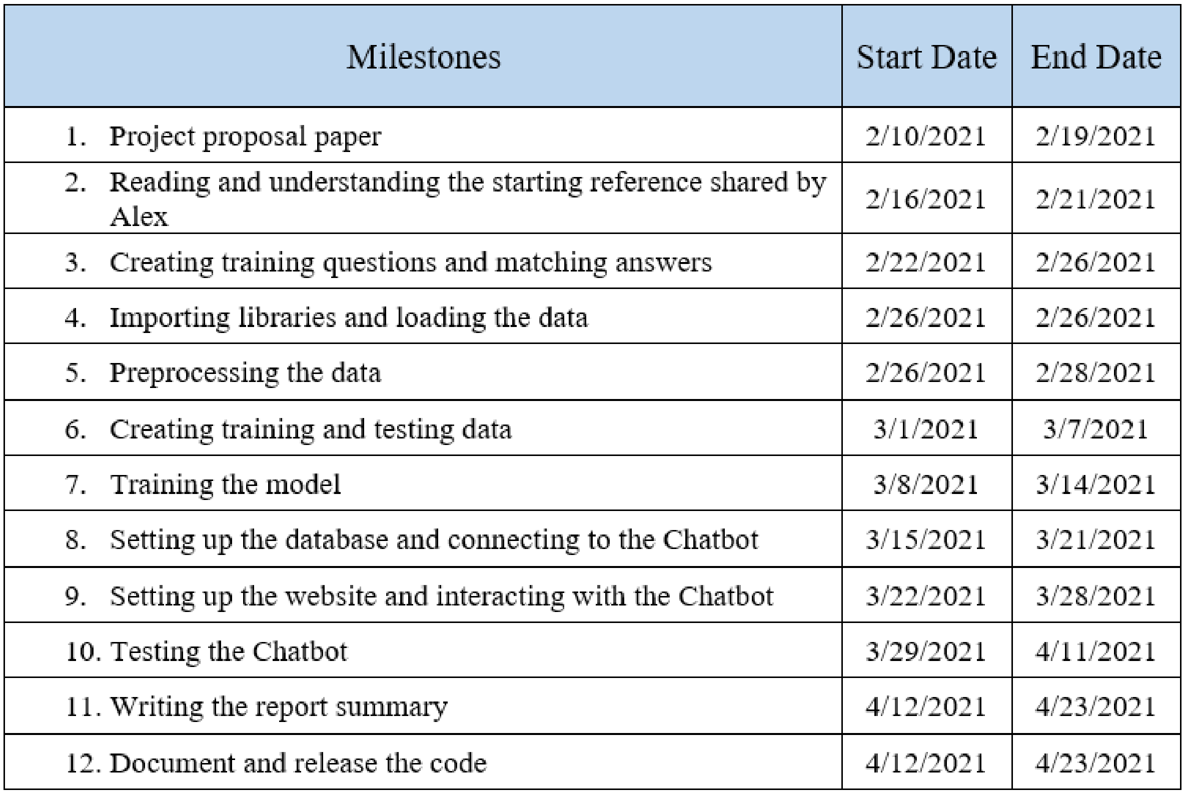
Framework

Building a conversational bot needs 3 big steps. We are planning to start with a chatbot based on manual templates so that the chatbot can react to some most basic conversations. For example, if the user enters “What is the full name of Kathy?”, the chatbot should respond with “It's the Cathedral of Learning!” This step should be the easiest to deal with, the chatbot will simply answer the questions they know word-by-word. The advantage of this method is accuracy, but the disadvantage is that it requires a lot of manpower, lacks flexibility, and has poor scalability.

After having a basic chatbot, our team will start working on how to improve it into a search-based chatbot. A search-based chatbot can fulfill the basic requirements of answering questions from the students. According to the sentence entered by the user, the chatbot will search and match the keywords in the library, then it will find the most suitable response content and return the answer to the user. This method has high requirements on the data library and needs it to be large enough, and its advantage is that the answers are high quality and the expression should be more natural.

After we got our search-based chatbot, our group will start our final improvement: a chatbot based on deep learning. A deep-learning chatbot will understand questions phrase-by-phrase based on the sentence entered by the user. It will then use the model we build to generate answers and reply to the user. Most of these technologies use the Encoder-Decoder model. (Encoders convert data into a required format. Decoders convert a coded message back to intelligible language) When implementing a chat system, there are many models we can find online, we will need to find the best that can fit our own chatbot. This method is scalable and can understand the context, but the model is difficult to train, and there are often a lot of grammatical mistakes when responding. I am not sure how far our group can go on the Deep-learning chatbot.

4. Project Execution



Milestones	Start Date	End Date
1. Project proposal paper	2/10/2021	2/19/2021
2. Reading and understanding the starting reference shared by Alex	2/16/2021	2/21/2021
3. Creating training questions and matching answers	2/22/2021	2/26/2021
4. Importing libraries and loading the data	2/26/2021	2/26/2021
5. Preprocessing the data	2/26/2021	2/28/2021
6. Creating training and testing data	3/1/2021	3/7/2021
7. Training the model	3/8/2021	3/14/2021
8. Setting up the database and connecting to the Chatbot	3/15/2021	3/21/2021
9. Setting up the website and interacting with the Chatbot	3/22/2021	3/28/2021
10. Testing the Chatbot	3/29/2021	4/11/2021
11. Writing the report summary	4/12/2021	4/23/2021
12. Document and release the code	4/12/2021	4/23/2021

5. Front-End Application Components

Website Structure

Virtual Kathy will be presented on a website within the pitt.cs domain so everybody from School of Computing and Information can use it. We plan on using the Flask framework (Python) to develop the main frame/structure of the webpage and will be taking full advantage of Flask's many addons such as Flask Bootstrap and Flask WTF Forms.

AI/ML Models and Model Training

After having the general structure of our Flask-framework based website in place, we will then develop our natural language processing model on Google Cloud Platforms (GCP) using Auto ML and data we gathered. Once we have the model ready, we will be testing it by making Python scripts that collect user inputs, then it's fairly easy to connect our Python scripts to our Flask Framework website. (Python scripts will take user inputs from our website's text entries)

Deployment

Once we complete Model Training, we will start testing the website on our local machine and debug the website. After that we will deploy our website on GCP. There are many popular options for deploying a website but our team member, George Liu, has the experience of deploying a website using GCP.

Community Testing and Maintenance

Once we deploy *Virtual Kathy* on GCP, we will distribute the link to our website to Pitt SCI community to begin the open testing phase and collect report logs (such as error logs, unexpected input logs etc.) Our team will continue to refine *Virtuala Kathy* based on these logs.

6. Project Organization

Team Meetings

Within our teams, we plan on using WeChat and GroupMe to stay in touch and keep up with the progress. These two platforms work the easiest for the team members since we all use them. To communicate with the project sponsor Alex, we would primarily use email, directly sent to Alex and his assistant, to report any issues we run into or any uncertainties we have. The team has also made an agreement to meet with Alex on a regular basis to discuss some major decisions (Alex proposed that we should meet once every three weeks at the least, but we expect to meet more often). Additionally, we are also available on Discord where we can communicate with Luis about any project-related issues.

Communication Languages

Since all our team members happen to be Chinese speakers, we will use both Chinese and English within our team. We figured that would be easier for us to communicate in Chinese among the group members for basic communications, but we will also be using English for more technical based discussions, as well as any communications with the sponsor and the instructor.

Mentor Meetings

To make sure that the team members and the project sponsor are on the same page, we will frequently update our works on the GitHub repository which is shared between Alex and the team. The team would start working on the project as soon as Alex shares the repository and some additional resources we need to get started on the project on GitHub. Therefore, we will use GitHub to share the product and our progress with the sponsor.

Sharing, Distribution and Archival of Project Data

All of the group work will be shared in a GitHub repository, and the URL for the project is provided below. The data files for our project will be saved both in team members' local machines and backed up in a Google drive remotely.
URL: https://github.com/PittCompSci/virtual-kathy-v1

7. References

[1] Adam, M., Wessel, M. & Benlian, A. AI-based chatbots in customer service and their effects on user compliance. Electron Markets (2020). https://doi.org/10.1007/s12525-020-00414-7

[2] Cahn, B.J. CHATBOT: Architecture, Design, & Development. Master's Thesis, University of Pennsylvania School of Engineering and Applied Science Department of Computer and Information Science, Philadelphia, PA, USA, 2017.

[3] Rahman, A.M.; Al Mamun, A.; Islam, A. Programming challenges of chatbot: Current and future prospective. In Proceedings of the Humanitarian Technology Conference (R10-HTC), Rajshahi, Bangladesh, 23 January 2018.