**X-o-Bot: A Generic Dialog Framework for Conversational Agents**

**Group Name** - SeungBot

**Members:**

Subham Anand

Z5151878

s.anand@student.unsw.edu.au

**Role** : Scrum Master and Developer

Vivek Arunagiri

Z5141013

v.arunagiri@student.unsw.edu.au

**Role** : Developer

Jiaying Guo

Z5129971

jiaying.guo@student.unsw.edu.au

**Role** : Developer

SheHoi Yeung

Z5130257

z5130257@student.unsw.edu.au

**Role** : Developer

**Proposal Submission Date** : 08.03.2019

**BackGround**

*A* ***chatbot*** *(also known as a smartbot, talkbot, chatterbot, Bot, IM bot, interactive agent, conversational interface, Conversational AI, or artificial conversational entity) is a computer program or an artificial intelligence which conducts a conversation via auditory or textual methods.*

Chatbot first introduced in 1966 - Eliza which used to the mimicked human conversation by matching user prompts to scripted responses - it was able, at least for a time, to pass the Turing artificial intelligence test. From there it continues to evolve with time and we saw some chatbots like PARRY, ALICE, SIRI, ALEXA, GOOGLE Assistant and CORTANA.

Chatbots are introduced in many fields nowadays and one of them is education which helps students and other professionals as well. Bostify is an education chatbot which gives a specific topic to students in the form of text, images, videos. After learning the topic students take quizzes, assignments and submit the results to their teachers. This way teachers can easily track students’ performance.

In universities, help desks are set up to solve or answer any small queries of students which can be answered using chatbots. So we decided our topic to be a student - tutor support system which will extract all the content from the specific course and answering the key questions related to the course.

In the existing system, the answers given by the chatbots are not accurate rather generalised ones which create ambiguity. The response time of chatbots is quite high and not instant. Integration of the chatbot with the different application is not yet configured.

**Aim**

After having a rich brainstorm session with our team we set our goal to create a simple user interface which will answer all the possible questions a student can ask the tutor regarding a particular course. The answers provided by the chatbot will be precise and reliable. The response time will be high and the user does not have to wait to get the reply. The chatbot can be plugged into multiple application and used as enquiry assistant.

1. Content related information.
2. The chatbot will provide answers related to the content of the course.
3. Learning outcome from the course.
4. Overview of the course.

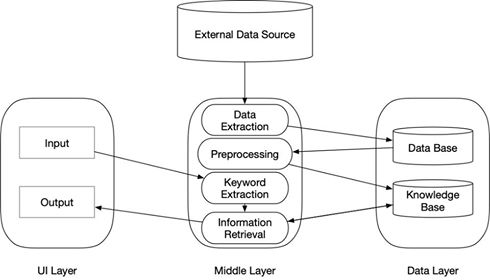
2. Lecture related information.

1. Class time table and professor details.
2. Lecture and lab location details.

3. Assignment and exam related information.

1. Questions related to assignments, quizzes and projects.
2. Questions related to the mid-term exam and the final exam.

**Design and Architecture**

****

Our system design consists of three layers UI layer, Middle layer and Data layer.

UI layer interacts with the user by getting the input question passing it to the middle layer and getting the response from it and showing it to the user. The visual look of the UI layer is simple with just three things input, output box and a button.

Middle layer performs all the operations to provide the answers with high accuracy and precision. It starts with data extraction from different sources such as previous year forum questions of the course and collection of question answers from StackOverflow and different other sources to train our model with more questions. To get more accuracy we need to preprocess the data which includes data cleaning and removing the noises from the data. When a user asks a question to the system we need to extract keywords from the question and send it to the data layer to get the response. Information retrieval is the process to get back answer from our corpus in the database to provide it to the user.

Data layer will consist of two types of data corpus-based and knowledge-based. All the data will be preprocessed and clean already.

**Scope**

We have decided to build a chatbot that can be attached to any existing chat service. We will be building a chatbot that simulates the interaction between a student and a tutor, where the student has questions about the details and structure of a course offered by UNSW and the chatbot can answer the question or link the student to the appropriate page.

We will be using COMP9444 - Neural Networks and Deep Learning to show how our system would work, but this can be easily modified for any other course. We intend to extract all the relevant information from the **course website** and also all the questions and answers in the **forums**, we will also extract data from **StackOverflow** and store them in a cloud-based database. We will then convert this data into a knowledge base we can use to answer questions answered by the student. Based on the question asked by the student we will use NLP and ML algorithms to provide the most accurate answer.

For this, we will first need to build a web-scraper using python (BeautifulSoup) to extract all the relevant information from the website. Then we will store these in a cloud database (Amazon RDS). We will then have an interface (UI) where we plug in this API, where a student can ask a question, upon which we will extract the key terms from the question and using NLP and ML algorithms, we will match them to the appropriate answer and return that to the UI screen for the user to see. We will be also implementing a feedback system where the student can provide feedback saying whether the answer was helpful and this will be used to tune the algorithm.

**Technical Skill Set**

Language : Python, C#, C, C++, MySQL

Libraries : NumPy, SciPy, Pandas, Scikit-learn, OpenCV, Tensorflow

Strengths : Organisation, Teamwork, Communication

Experience : .Net developer, Software Engineering

**Project Choice**

Our choice of the chatbot project was made due to all our interests and curiosity in one of the more popular applications of Artificial Intelligence, the chatbots. Also, the decision to choose this project was due to the immense benefit it could have for students and the thought of how much it would have benefited me in my courses if I had a chatbot that could answer questions instantaneously.

**Epics**

We have split our epics based on the different functions of our Chatbot.

### **Data Extraction**

Data Extraction is to get the data from the external source. In this project, the course COMP9444-Neural Networks and Deep Learning content and detail as well as information from Stackoverflow website. First, we need to build a database to store the data which mapped from the external source. This function is a process of retrieving data out of data sources for further data processing or data storage. Web data extraction is the main function in the system.

For course website, we extract details of course, such as offered semester, course content, credit, prerequisite, timetable, course outline, study level, contact, class location and so on.

For StackOverflow, this project would be using machine learning to extract about 1000 question about basic computer science questions.

Example data extraction form:

|  |  |
| --- | --- |
| **Course Code** | COMP9444 |
| **Course Title** | Neural Networks and Deep Learning |
| **Convenor** | Alan Blair |
| **Lectures** | Monday 6-9pm in Central Lecture Block 7, Weeks 1-9,11-13 |
| **Prerequisites** | COMP1927 or COMP9024 |
| **...** | ... |

**Query Classification**

Our problem domain revolves around types of questions asked by the users. Here the interaction happens between tutor and students where students will ask questions related to the course in which he is enrolled. The questions can be of different types which include content, lectures, assignments and exams. It is important to classify the question types so that it will be easier for our model to get answers from the database and knowledge base.

We get questions from different sources like forum page of the course, StackOverflow and a few other sources. We then label the data or questions as what types they fall into so that we can quantify the performance of the model.

Query classification revolves around the use of keyword extraction from the questions and divides them into appropriate classes.

Example of types of questions user can ask for the course comp9444 and the categories they fall into.

|  |  |  |
| --- | --- | --- |
| Comp9444 - Neural Network and Deep learning | Query | Category/Classes |
| 1 | Which book is recommended to use in this course? | Content related query |
| 2 | Where can I find the lecture slides and recording? | Lecture related query |
| 3 | When is the deadline for assignment one? | Assignment related query |

**Information Retrieval**

This module is a fundamental epic for our application. When a user asks the question in the UI layer, the keywords from the question are first extracted using NLP algorithms. This will include pre-processing the question to remove stop words and extract only the relevant keywords.

These keywords will be used by our Machine Learning algorithms to find the question-answer pair in the knowledge base that best matches the question put forth by the student. This answer is then returned to the UI layer so that it can be displayed to the user.

Example:

Q: Hi! Quick Question about how our networks are going to be tested.

Are they going to be run up to a certain number of episodes? The spec says that all runs will timeout after 2mins (regardless of what ep they reach) and that any learning after 500eps will reduce our mark to 5.

A: See the "Marks" section in the spec. In short, you want to have the env solved after 100 episodes (hint: this cannot be done with online learning - requires batching + exp. replay), and you want learning to be monotonic and stable for full marks.

**Stand-up/Sprint Schedule**

Our team will be using the agile methodologies, with weekly sprints and stand-ups throughout the week. We will also track our sprint on a Trello board and commit our code to the GitHub repository that we have set up.

**Stand-up:**

* Monday (Online)
* Thursday (In person)

**Sprints:**

* Every Week (Week 4 - Week 9)

**Weekly Meeting Times**

* Thursday, Bongo Lab K17, 12:00 - 14:00

**References**

*MOOC-O-Bot: Using Cognitive Technologies to Extend Knowledge Support in MOOCs*

*https://en.wikipedia.org/wiki/Chatbot*

*https://pcc.cs.byu.edu/2018/03/26/a-brief-history-of-chatbots/*