**Intelligent myCourseBot using advanced ML algorithm and Python**

**Group Name:** Py Developers

**Group Members:**

|  |  |  |
| --- | --- | --- |
| **First name** | **Last Name** | **Student number** |
| **Patricia** | **Adolph** |  |
| **Lekshmi Chandran** | **Sheela** |  |
| **Pooja** | **Selby** |  |

**Submission date: *16/08/2021***

Contents

[Abstract 3](file:///C:\Users\Lekshmi\Downloads\AML-1204-Project-Report-Template.docx#_Toc4571289)

[Introduction 3](file:///C:\Users\Lekshmi\Downloads\AML-1204-Project-Report-Template.docx#_Toc4571290)

[Methods 3](file:///C:\Users\Lekshmi\Downloads\AML-1204-Project-Report-Template.docx#_Toc4571291)

[Results 3](file:///C:\Users\Lekshmi\Downloads\AML-1204-Project-Report-Template.docx#_Toc4571292)

[Conclusions and Future Work 4](file:///C:\Users\Lekshmi\Downloads\AML-1204-Project-Report-Template.docx#_Toc4571293)

[References 4](file:///C:\Users\Lekshmi\Downloads\AML-1204-Project-Report-Template.docx#_Toc4571294)

**Abstract**

Our objective was to build a knowledge system using a chatbot exclusively for AI and ML course. The name of our chatbot is “myCourseBot”, which can be used as a virtual assistant for solving student queries related to AI and ML course. In the initial stage, we are trying to build a small prototype using a limited AI and ML knowledge base as a data source that can be leveraged in the future. We can enhance the knowledge base by training our chatbot with AI and ML related information collected from other sources, students from our class, and upcoming batches. This bot can be used as AI and ML course assistance. The knowledge system of the bot will be trained using advanced machine learning algorithms like deep learning.

**Introduction**

A chatbot is a computer (AI) software that imitate and processes human communication (written or spoken), allowing people to connect with digital gadgets as if they were conversing with actual people. Chatbots may be as simple as one-line programmed that respond to a simple question, or as complex as digital assistants that learn and grow as they gather and analyses data to offer increasing degrees of customization. Chatbots process data to respond to a variety of queries, using AI, natural-language processing (NLP), automated rules and machine learning (ML).

Our project is to create a knowledge system using chatbot specifically for our AI and ML course. Using this chatbot any student can solve queries related to AI and ML course. The name of the chatbot is “myCourseBot”. This Bot is a web-based application with a front end (using Angular) and backend (using Flask). In the GUI the user can input his/her queries and finally it will show responses with the help of ML and backend.

**Methods**

Our project has 5 sections,

1. Storing the data in MySQL
2. Data Transformation
3. Chatbot ML Model Training
4. Predict the Responses
5. Giving Responses to end user through Angular

The following are a detailed description of the modules:

1. **Storing the data in MySQL database.**

We have used MySQL as the database for storing the intents in a structured format. There are three fields – tag, patterns, and responses where ‘Tag’ is the category in each intent belongs to, ‘Patterns’ are the queries that the user will likely ask to the chatbot, and ‘Response’ is the responses that a user should get. The name of the database is ‘chatbot\_db\_test’, and the table name is ‘chatbot\_tb’

Text

Description automatically generated with medium confidence

Figure 1: Structure of Chatbot Database

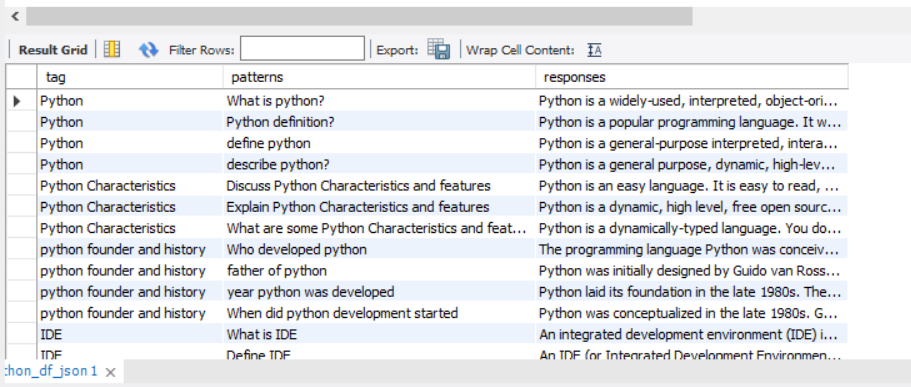


Figure 2: Chatbot Table

1. **Data Transformation – Data.py:**

Before modeling we have performed some pre-processing on the structured dataset to convert it into json format. Firstly a connection is established with the MySQL database and all the records and retrieved from the database and is saved as a dataframe using Pandas library. The data is finally saved as a json file after the pre-processing.

Text

Description automatically generated

Figure 3: Connection between Python and MySQL

1. **Chatbot ML Model Training – Train.py**

The data saved as a pre-defined json file will then be passed on to ‘train.py’ file where the data is lemmatized, tokenized, bag of words created and they are stored as classes, words and documents as separate lists and saved as pickle files.

Tokenising is the first thing to do on the text data. Tokenising is the process through which we break the whole sentence into subparts like words. We iterate through each pattern in intents and then tokenize the sentences using ‘nltk.word\_tokenise’ function. These tokenised words are then appended to a list word. The tokenised words and the tags are appended to a document list and a list of classes are also appended with the tags.

The next step is to lemmatize each word which means to convert a word to it’s base form. For eg: Lemmatization of ‘Caring’ is care. After converting words into lemma form, we created a pickle file for storing objects that can further be used for predicting. So, two pickle files are used to store words and classes.

A deep neural network machine learning model using tensorflow is created to train the chatbot. The chatbot model is a sequential model. It has three layers were the first layer contains 128 neurons, second layer 64 neurons and the third output layer has number of neurons equal to number of intents to predict the output intent with the activation function ‘softmax’. The next step is to compile the model, Stochastic gradient descent with Nesterov accelerated gradient gave good results for our model. It is a method which finds the optimal parameters configuration for algorithms by adjusting the configurations of the network after each training point. After training the model with 200 epochs, we got an accuracy 77% and saved the model as ‘chatbot\_model.h5’ file. We can improve the accuracy if more training data is being added to the database.

1. **Predict the Response – App.py**

To predict the sentence and get accurate responses we have created App.py. First install all the necessary libraries like flask, nltk, json, pickle etc.

Initially we will load the trained model that is ‘chatbot\_model.h5’. We have used Flask framework to integrate the graphical user interface (Angular) and API connection which helps to build web applications. Following are the procedure.

* First, we created a flask app to host the application using:

app = Flask(\_\_name\_\_)

* Creates a decorator route function that calls the API which extracts chats from the front end and passing it in the trained model in order to provide prompt responses to the users.
* When the development server is up and running in the browser, we can do the actions accordingly.

To identify the class and to retrieve random response from the list of responses we have used some functions in ‘app.py’. These functions will first perform preprocessing of text to predict the class.

We used tokenizing and lemmatization for text preprocessing and returns cleaned up sentences which is stored in a bag of words array. If word in the bag exists in the sentence it is represented as 0 or 1. The next step is to tokenize the pattern.

The ‘predict\_class’ function is used to filter out predictions below a threshold, the error\_threshold that we assigned is 0.25, the prediction of class will return random responses from the list of intents which will be returned as responses to the end-user.

Steps to run the chat bot:

1. Create connection between Sql and python by running data.py
2. Train the model by running train.py
3. Establish connection between Flask and Angular by running App.py.
4. To run angular, open command prompt from the folder where we have the angular code and run npm install after that run ng serve which will successfully run the application in the browser.
5. **Front End using Angular**

The frontend allows the users to ask queries related to AI and ML course.

We used angular for making the frontend. First, we installed node and angular cli. Then create a simple angular app using angular cli.

Installed packages for angular applications are:

* Angular CLI: 12.2.1
* Node: 16.6.1
* Package Manager: npm 7.20.6

Package Version

---------------------------------------------------------

@angular-devkit/architect 0.1202.1

@angular-devkit/build-angular 12.2.1

@angular-devkit/core 12.2.1

@angular-devkit/schematics 12.2.1 (cli-only)

@schematics/angular 12.2.1 (cli-only)   
  
ng new angular-chat-bot  
cd angular-chat-bot  
ng serve

Then created folder called components and edited app.component.ts inside folder chatbot like below:

Text

Description automatically generated

Figure 4: Chatbot Component

It will get the queries from the user and in return give responses as per the intents added by us.

**Result**

We have successfully integrated version1 of our ‘MyCourseBot’ which has MySQL database, chatbot model and angular front end. We have tested our model and it gave us satisfactory results. We are working on Version of our chatbot model as a future work.

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure 5: MyCourseBot GUI

**Future Work**

We have developed a prototype of myCourseBot version 2 with some enhancements. The main highlight of this version would be that we can retrain our model with the new set of data if not already trained, and this will increase the knowledge base for our chatbot. The model will accept the intents in the format of a csv file which will be trained and appended into the exisisting database. As of now, the process is not automated, and this can be leveraged in the future. The myCourseBot can be expanded by including all the subjects for each semester in AI and ML course. So, in future this bot can be implemented in Lambton to be used as a course assistant for the course Artificial Intelligence and Machine learning.

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure 6: Prototype of version 2 Mycourse Bot

**Conclusion**

It is very common that students at all grades are interested to talk with teachers through messages. So, the course bot will provide assistance to students who are very new to the course, and they can easily grab the basic information from the bot. Our bot can be leveraged in the future and can be replicated in large scale, through which students get the advantages to ask questions and get quick responses. Personalized learning can be achieved through our chat-bot.

**References**

Kégl, B. (2016, May 31). *The data science ecosystem: industrial edition*. Towards Datascience. <https://towardsdatascience.com/the-data-science-ecosystem-industrial-edition-938582427466>.

*What is flask python*. What is Flask Python - Python Tutorial. (n.d.). <https://pythonbasics.org/what-is-flask-python/>.

# *Python Chatbot Project – Learn to build your first chatbot using NLTK & Keras*. Data Flair. (n.d).

<https://data-flair.training/blogs/python-chatbot-project/>

*Angularjs application*. Angular Application. (n.d.). <https://www.w3schools.com/angular/angular_application.asp>.

**Appendix**

MyCourseBot workings:

****