CMS Made Easy…Automated Chatbot for Students

FYP Proposal Number
CS-FYP-2018-001
Date of Submission
12-Jul-2020

Students:
Balaj Yousuf / BSCS/F15/0133
Saad Nasir / BSCS/F15/0102
Usama Amjad / BSCS/F15/0116

Supervisor
Asst. Prof Imran Khan

Co-Supervisor
Asst. Prof Iqbal Uddin Khan

Department of Computing
Faculty of Engineering & Science
Hamdard University

Contents

Table of Figures

1.0 Introduction

1.1 Problem Statement

In this technologically advanced world, navigating between pages or to perform queries in every other page wastes a lot of time. People now a days prefer those applications where they can see everything in just one place. Stereotype websites are being replaced by Single Page Applications and Progressive Web Apps.

1.2 Motivation

From my personal experience, navigating between CMS takes a lot of time. For e.g. just to see my current schedule or transport or tuition fee we have to navigate back and forth on average 4 times. In the age where 5G is evolving the world, this is a no brainer. Chatbots are one of many solutions that can solve this problem. You just ask chatbot the question, it will provide with you the relevant answers.

1.3 Objective

The objective is to develop a generalized algorithm that is able to detect what user wants to know with the help of Artificial Intelligence and provide appropriate answer. The Aim is to develop a model that can be integrated with our own university CMS and help students serve better way.

2.0 Project Scope

It centers on the ability to help user create an interactive way to use CMS. Chatbot will provide a unique way of seeing data and visualizing with CMS. It will make user fun to open CMS.

We will train our machine learning models until they are fitting enough to be used on application. Our project will mainly hub on developing a machine learning model and training it to accomplish high level of accuracy so it can be used for other research-oriented projects and can be further specified according to the need of the organization.

Our basic endeavor is to train model using Generative approach and generative models on the seq2seq neural network. This network was initially released for machine translation, but has also proved to be quite effective when it comes to building generative Chatbots . Generative Chatbots also require a very large amount of conversational data to train. We will first train our seq2seq implementation for our chatbot using more 2 million conversations. also require a very large amount of conversational data to train. If we think that Generative approach is not working according to our need, we will switch to Retrieval Based bots. Retrieval based bots work on the principle of directed flows or graphs. The bot is trained to rank the best response from a finite set of predefined responses.

3.0 Progress Evaluation

First, we created our own Dataset of Students query which they would normally use if they wish to communicate with the bot. We also developed a dummy database which will act as our CMS to which our chatbot will be integrated. The database is a SQL based Database. We have then trained our dataset using come machine learning models which are used for Natural Language Processing. More Details are provided in headings below

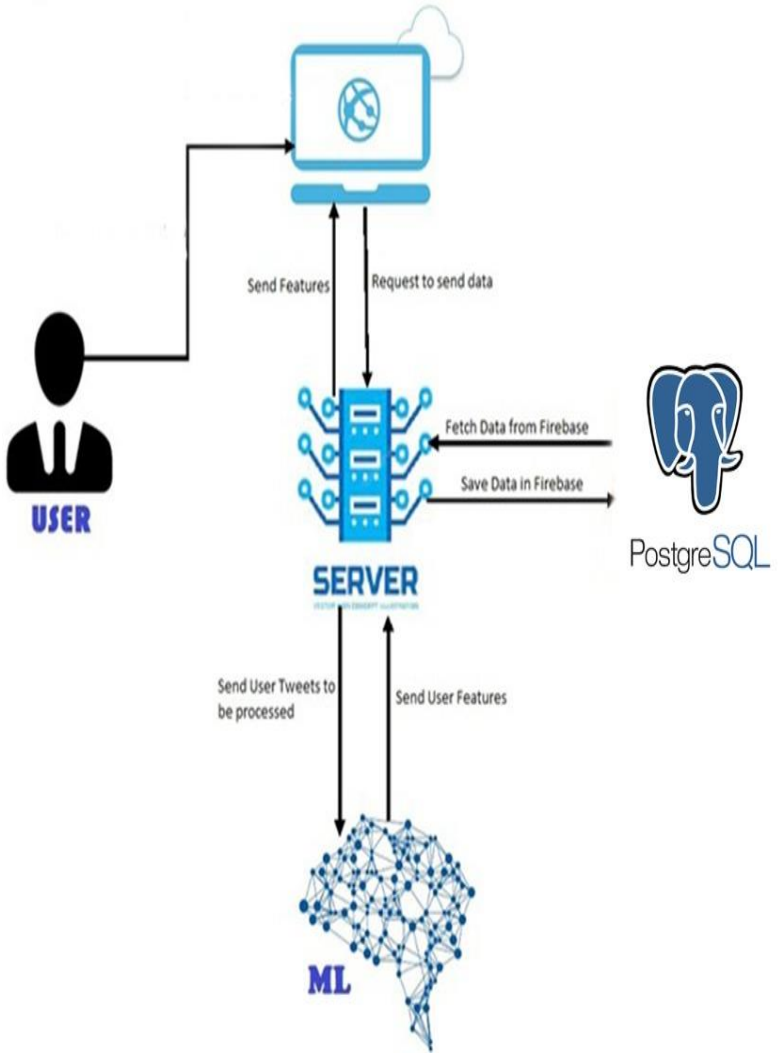


Fig 1: Front-end, Back-end, Database Integration

3.1 Generating Datasets

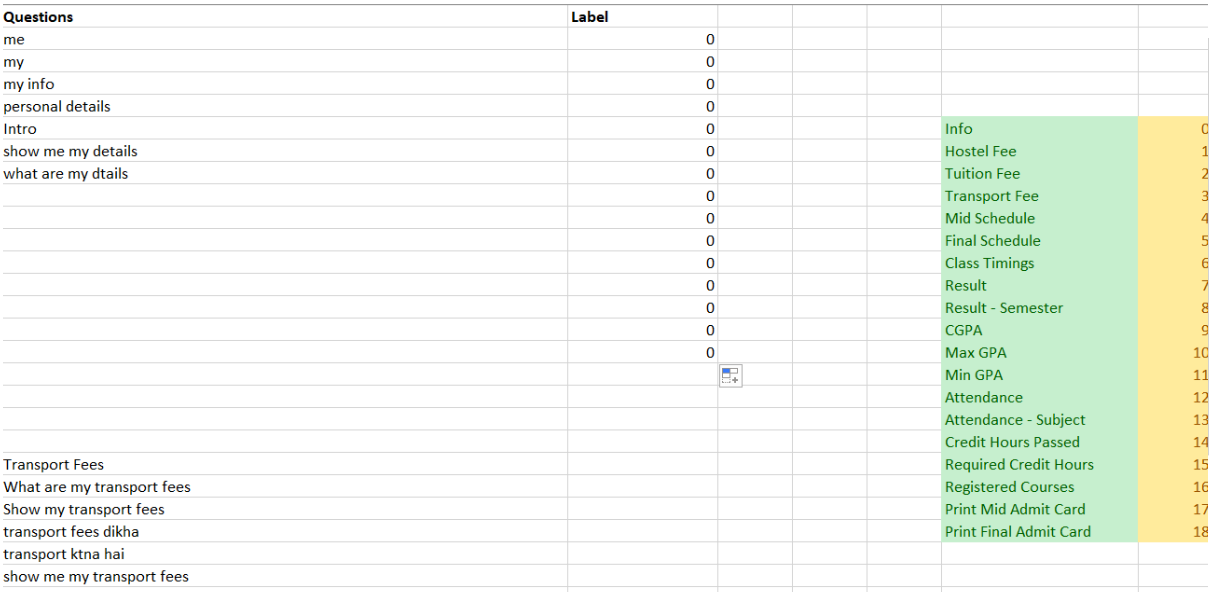
We have not used any previously available datasets on Internet because we wanted to make our own customized dataset which enables us to give better insights about what category student / user might query if they were to communicate with a bot. The dataset is a carefully studied dataset by studying students' traits and by preparing some survey forms and analyzing it. Overall using this technique, we were able to gather more than 150,000 rows for our dataset.

3.2 Data Cleaning

After collecting data, we needed to prepare data to convert it into meaningful information so it can be used to train our model. To clean our data we have used following techniques

1. Applying lemmatization on words to make it in pure English Language form. Lemmatization removes all suffixes from words and converts it to its present state.
2. Remove all stopwords because they are of not much information. Stopwords are 'in', 'am', 'ok', 'is' etc.
3. Convert all queries to lower case to remove any case sensitivity issue.
4. Strip all queries from extra white spaces.
5. Remove all words with length less than 2 because they are also counted as additional words.

After applying all these techniques this is how our dataset looks.

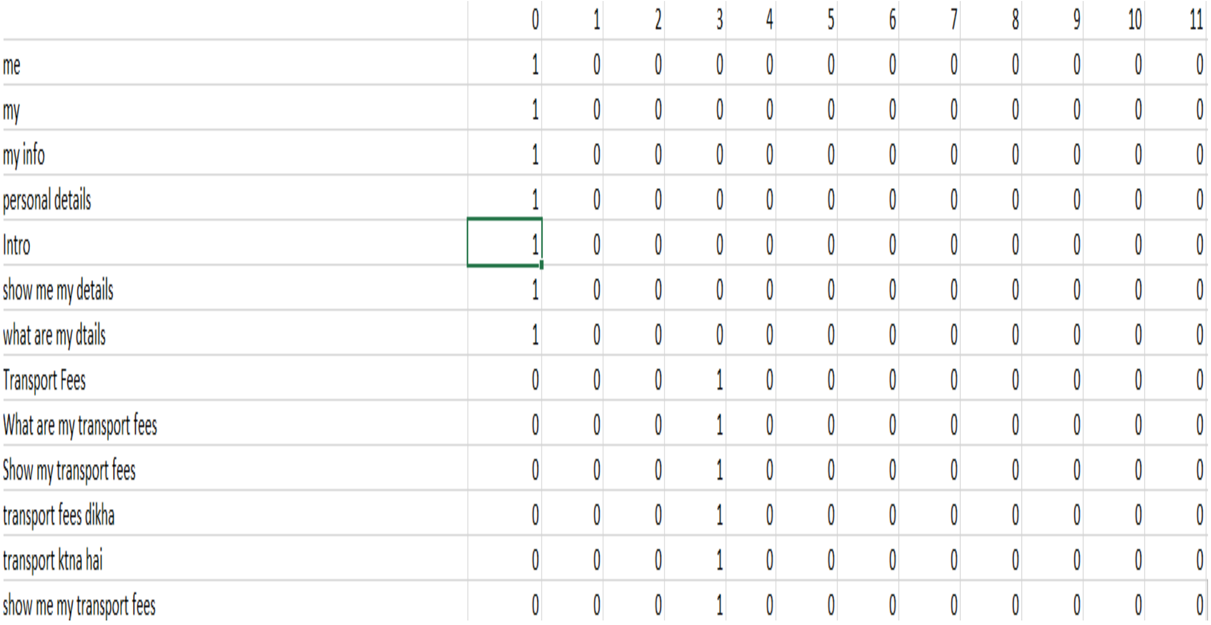


Questions	Label		
me	0		
my	0		
my info	0		
personal details	0		
Intro	0	Info	0
show me my details	0	Hostel Fee	1
what are my dtails	0	Tuition Fee	2
	0	Transport Fee	3
	0	Mid Schedule	4
	0	Final Schedule	5
	0	Class Timings	6
	0	Result	7
	0	Result - Semester	8
	0	CGPA	9
	0	Max GPA	10
		Min GPA	11
		Attendance	12
		Attendance - Subject	13
		Credit Hours Passed	14
Transport Fees		Required Credit Hours	15
What are my transport fees		Registered Courses	16
Show my transport fees		Print Mid Admit Card	17
transport fees dikha		Print Final Admit Card	18
transport ktna hai			
show me my transport fees			

Fig 2: Queries dataset after doing data cleaning

3.3 Developing Keywords file

We made a manual keywords of all categories and give each word its due weightage according to the words efficiency as we thought it has. Then we matched those keywords with our Queries dataset and give each query its weightage for each category and the category which has highest score will be assigned that category. Applying this keyword matching enabled us to Label our Queries. This is how then our Queries dataset look.



	0	1	2	3	4	5	6	7	8	9	10	11
me	1	0	0	0	0	0	0	0	0	0	0	0
my	1	0	0	0	0	0	0	0	0	0	0	0
my info	1	0	0	0	0	0	0	0	0	0	0	0
personal details	1	0	0	0	0	0	0	0	0	0	0	0
Intro	1	0	0	0	0	0	0	0	0	0	0	0
show me my details	1	0	0	0	0	0	0	0	0	0	0	0
what are my dtails	1	0	0	0	0	0	0	0	0	0	0	0
Transport Fees	0	0	0	1	0	0	0	0	0	0	0	0
What are my transport fees	0	0	0	1	0	0	0	0	0	0	0	0
Show my transport fees	0	0	0	1	0	0	0	0	0	0	0	0
transport fees dikha	0	0	0	1	0	0	0	0	0	0	0	0
transport ktna hai	0	0	0	1	0	0	0	0	0	0	0	0
show me my transport fees	0	0	0	1	0	0	0	0	0	0	0	0

Fig 3: Queries dataset after Labelling

3.4 Generating Keyword File

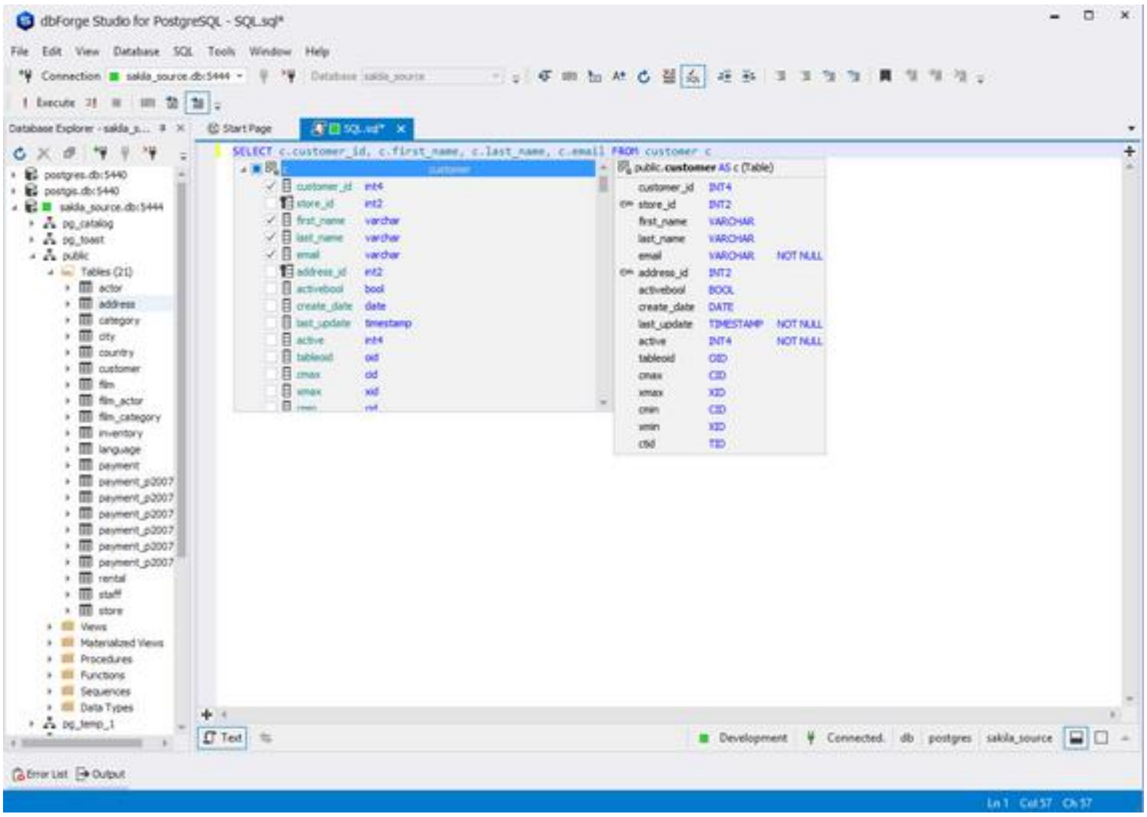
After Labelling all queries we will apply TFIDF Vectorizer using Scikit library in python, TFIDF Vectorizer works in a way that it determines frequency of all words in our corpus and then applies inverse document Frequency technique on them which searched for least used words and in how many queries those word is occurring. Using this technique we will again generate a keyword file which will be generated by scikit libraries function 'get features'. These keyword file will also be in JSON format with keywords as its 'keys' and weightages as its 'values'.

3.5 Training our Model using Different Machine Learning Algorithms

Now after creating the keywords file we trained our datasets using different machine learning models and see its accuracy and efficiency. This process is still in continuous state as we explore to increase its accuracy to ~95%

3.6 Database

We have used Postgres as our SQL Database.



```sql
SELECT c.customer_id, c.first_name, c.last_name, c.email FROM customer c
```

customer

✓ customer_id	int4
store_id	int2
✓ first_name	varchar
✓ last_name	varchar
✓ email	varchar
address_id	int2
activebool	bool
create_date	date
last_update	timestamp
active	int4
tableoid	oid
cmax	cid
xmax	xid

public.**customer** AS c (Table)

customer_id	INT4	
store_id	INT2	
first_name	VARCHAR	
last_name	VARCHAR	
email	VARCHAR	NOT NULL
address_id	INT2	
activebool	BOOL	
create_date	DATE	
last_update	TIMESTAMP	NOT NULL
active	INT4	NOT NULL
tableoid	OID	
cmax	CID	
xmax	XID	
cmin	CID	
xmin	XID	
ctid	TID	

Fig 5: Postgres as our Database

4.0 Process Flow

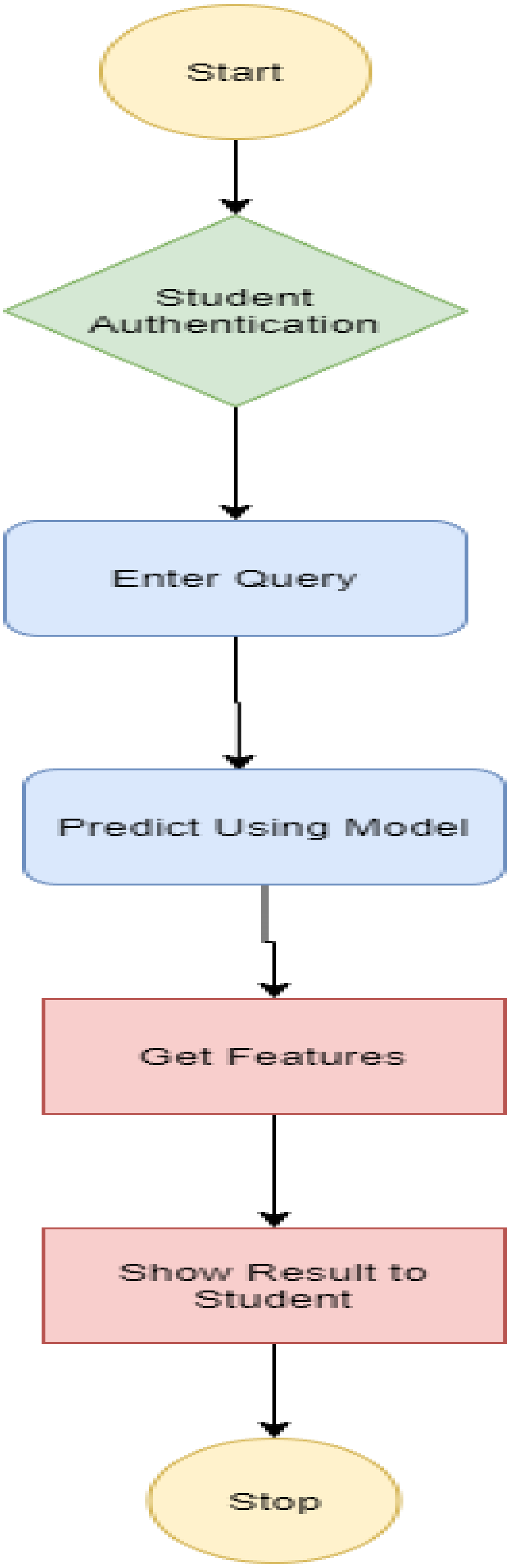


Fig 6: Work Flow Diagram of User/Admin

5.0 Milestones

- **At the end of 8th semester (on final evaluation):**
  In our Final Evaluation we will give following objects.
  1) Complete Web Application with front-end and backend both integrated together with database.
  2) A Research Paper
  3) A Machine Learning Model Trained according to our needs with higher accuracy
  4) All documents with complete documentation.