Chattberbot,
A chatbot that answers queries about VIT.
Artificial Intelligence, 2019

INDEX

ABSTRACT

With the ever-increasing growth in communication technology, comes the idea of chatter robots or chatbots. A chatbot is a computer program which can conduct a conversation through auditory or textual methods. They are often designed to convincingly simulate how a human would behave as a conversational partner. Chatbots are typically used in dialog systems for various practical purposes such as customer service or information acquisition. Where Chatbots are used:

1) Messaging Platforms: Chatbots are used as a part of instant messaging platforms like Facebook messenger, WeChat for entertaining reasons as well as customer service, sales and marketing.

2) In apps and on websites: Previous generations of Chatbots were present on the company websites, the new generation of Chatbots now assist which includes the IBM Watson-powered “Rocky”.

3) Company internal Platforms: Companies use Chatbots internally for customer support, Human Resources, etc.

4) Toys: Chatbots have also been incorporated into devices not primarily meant for computing such as toys. Chatbot creation follows a pattern similar to the development of a web page or a mobile app.

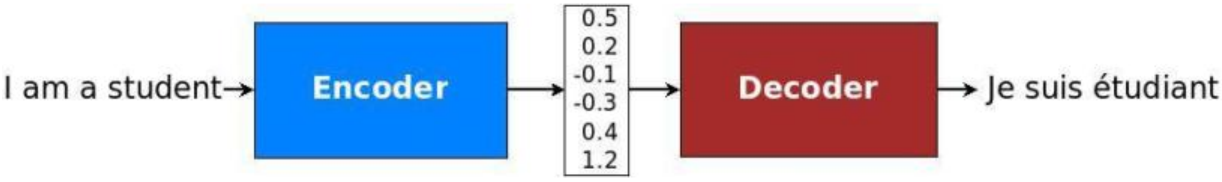
1. INTRODUCTION

A Chatbot ("Chatter Robot") is a computer program that mimics human conversations in its natural format including text or spoken language using artificial intelligence techniques such as Natural Language Processing (NLP), image and video processing, and audio analysis. The most interesting feature of the bots is that they learn from the past interactions and become intelligent and smarter over the time. Our project acutely deals with an important section of this growing entity, focusing the usage of the chatbot sin the field of entertainment like Google Now, Apple Siri, etc.

The entire project will be implemented in Python using Googles Deep Learning library TensorFlow. TensorFlow comes with a lot of different neural networks, the one which we will be working is Recurrent Neural Network. Neural Machine Translation is the core concept used by bots. Back in the old days, traditional phrase-based translation systems performed their task by breaking up source sentences into multiple chunks and then translated them phrase-by-phrase. This led to disfluency in the translation outputs and was not quite like how we, humans, translate. We read the entire source sentence, understand its meaning, and then produce a translation. Neural Machine Translation (NMT) mimics that.

Our goal is to create a chatbot that could talk to people on the Twitch Stream in real-time, and talk to people like a human being. The chatbot will be trained on the Reddit Comments Dataset and it will be trained on a Recurrent Neural Network. The chatbot will be able to talk to people and the main objective will be to get the bot to answer trivial questions asked by the user. Users can chat using any format there is no specific format the user has to follow. The System uses built in artificial intelligence to answer the query. The answers are appropriate what the user queries. Abstract Chatbots use a database of responses often culled from a corpus of text generated for a different purpose, for example film scripts or interviews. The basic aim of this system is to bridge the vocabulary gap between people.

NMT models vary in terms of their exact architectures. A natural choice for sequential data is the recurrent neural network (RNN), used by most NMT models. Usually an RNN is used for both the encoder and decoder. The RNN models, however, differ in terms of: (a) *directionality* – unidirectional or bidirectional; (b) *depth* – single- or multi-layer; and (c) *type*– often either a vanilla RNN, a Long Short-term Memory (LSTM), or a gated recurrent unit (GRU).



Chatbot creation follows a pattern similar to the development of a web page or a mobile app.

Seq2Seq Model:

It consists of two recurrent neural networks (RNNs): an encoder that processes the input and a decoder that generates the output. This basic architecture is depicted below. Each box in the picture above represents a cell of the RNN, Encoder and decoder can share weights or use a different set of parameters. Multi-layer cells have been successfully used in sequence-to-sequence models. In the basic model depicted above, every input has to be encoded into a fixed-size state vector, as that is the only thing passed to the decoder.

The main idea is using the same seq2seq model as a language model, to get the candidate words with high probability in each decoding timestamp as a anti-model, then we penalize these words always being high probability for any input. By this anti-model, we could get more special, non-generic, informative response.

2. LITERATURE SURVERY

Serial Number	Title of the paper	Authors	Year of publica tion	Dataset used	Methodology/ Technology	Performance Metrics	Advantages	Disadvantages
1	Generative Indonesian Conversation Model using Recurrent Neural Network with Attention Mechanism	Andry Chowanda, Alan Darmasaputra Chowanda	2018	The largest number of dialogue collected belong to Twitter Conversation Triple Dataset, with more than 70M dialogues collected from Twitter conversation in English. Most of dataset collected were from Human-Human short conversation (i.e. chat from messenger or twitter).	The deep learning method used was Long-Short Term Memory algorithm, one of Recurrent Neural Network (RNN) algorithm.	vocabulary *size* = 24000. Resulted in 2.37% unknown word, 1.66 the rate of loss, and the perplexity of 4.96	Attention Mechanism algorithm provides better results compare to the one with-out the algorithm	Further research direction is to collect more dataset from conversation, not only from the movie subtitles but also from natural conversation between human and human in Indonesian Language.

Serial Number	Title of the paper	Authors	Year of publica tion	Dataset used	Methodology/ Technology	Performance Metrics	Advantages	Disadvantages
2	Implementation of A Neural Natural Language Understanding Component for Arabic Dialogue Systems Component for Arabic Dialogue Systems	Abdallah M. Bashir, Abubakr Hassan, Benjamin Rosman , Daniel Duma, ,Mohanad	2018	An online survey as there were no available sources for previous labeled datasets in Arabic oriented to Home Automation.	The Intent Classifier and the Entity Extractor. The Intent Classifier was implemented using two neural text classification techniques, which are LSTMs and CNNs. The Entity Extractor was implemented using a Bidirectional LSTM along with character-based word embeddings.	Both of the intent classification implementations were benchmarked and the results of the benchmark indicated that the LSTM performance with an F-Score	They used state-of-the-art neural network text classification techniques of CNN and LSTM to classify the user's intent.	Isn't integrated with Automatic Speech Recognition (ASR) and Natural Language Generation (NLG) modules to yield an efficient task oriented Dialogue System.

Serial Number	Title of the paper	Authors	Year of publication	Dataset used	Methodology/ Technology	Performance Metrics	Advantages	Disadvantages
3	A Deep Reinforcement Learning Chatbot	Iulian V. Serban, Chinnadhurai Sankar, Mathieu Germain, Saizheng Zhang, Zhouhan Lin, Sandeep Subramanian, Taesup Kim	2018	Amazon Mechanical Turk (AMT) to collect data for training the policy	First, it uses all response models to generate a set of candidate responses. Second, if there exists a priority response in the set of candidate responses (i.e. a response which takes precedence over other responses), this response will be returned by the system. Third, if there are no priority responses, the response is selected by the model selection policy.	The best performing system reached an average user score of3.15, on a scale1−5, with a minimal amount of hand-crafted states and rules and without engaging in non-conversational activities (such as playing games).	In comparison, the average user score is higher than most systems (abg-2.92). Furthermore, the same system averaged14.5−16.0turns per conversation, which is substantially higher than the average number of turns per conversation of all the existing systems. Their system is one of the most interactive and engaging systems.	The training set used heavily determines the extent of accuracy of the model.

Serial Number	Title of the paper	Authors	Year of publication	Dataset used	Methodology/ Technology	Performance Metrics	Advantages	Disadvantages
4	Intelligent Chatbot using Deep Learning	Anjana Tiha	2018	In this project, the dataset "Cornell Movie Subtitle Corpus" has been primarily used for final model training, along with "Open Movie Subtitle Corpus" and "Movie Subtitle Corpus" 225000*2 utterance of "Cornell movie subtitle corpus" conversation and has been tested with 5000+5000 utterance and validated with 5000+5000utterance	Developed an Intelligent conversational agent following state of the art techniques. Have used Google's NeuralMachine Translation(GNMT) module for building dialogue generator (GNMT used for dialogue generation and text summarization). GNMT has rich Seq2Seq module with many additional features for dialogue generation. GNMT also have option for NeuralAttention Mechanism,	Eval development, Perplexity: 50.76 and Bleu 10.1	It is a Sequence to Sequence module with encoder-decoder architecture built on bi directional LSTM cells and Neural attention mechanism, Beam Search and thus could be more preferred as the dialogue generation problem is solved.	As the model used in this experiment is for machine translation, the dialogue generation is treated as translation problem, where history of earlier conversations are not taken into account. Hence, the model can be limited in performance regarding long conversation

Serial Number	Title of the paper	Authors	Year of publication	Dataset used	Methodology/ Technology	Performance Metrics	Advantages	Disadvantages
5	Training Seq2Seq models together with Language Models	Anuroop, Sriram, HeewoJun, Sanjeev Satheesh, Adam Coates	2017	For each dataset, Amazon Mechanical Turk is used, to collect audio recordings of speakers reading out the text.	Seq2Seq hidden state and the language model hidden state are to be used as inputs to the gate computation.	For all models which were trained on the source domain, the source CER and WER indicate in-domain performance and the target CER and WER indicate out-of-domain performance.	Cold Fusion can narrow down the error rate to 18% while the other researches had 50% error rate.	First, decoder learns an implicit language model from the training labels, taking up a significant portion of the decoder capacity to learn redundant information. Second, the residual language model baked into the Seq2Seq decoder is biased towards the training labels of the parallel corpus.

Serial Number	Title of the paper	Authors	Year of publication	Dataset used	Methodology/ Technology	Performance Metrics	Advantages	Disadvantages
6	A Persona-Based Neural Conversation Model	JiweiLi Chris Brockett, Georgios P.Spithourakis	2016	Twitter Personal Dataset, Television Series Transcripts	This model encodes each individual speaker as a vector or embedding, which encodes speaker-specific information (eg dialect, register, age, gender) that influences the content and style of response.	BLEU Score which stands for Bilingual Evaluation Understudy Score.	Performance on the twitter persona dataset. Perplexity is reported in finding, observed about a 10% decrease in perplexity for the speaker model over the standard Seq2Seq model.	Propensity to select the response with greatest likelihood—in effect a consensus response of the humans represented in the training data.

Serial Number	Title of the paper	Authors	Year of publication	Dataset used	Methodology/ Technology	Performance Metrics	Advantages	Disadvantages
7	Deep Reinforcement Learning for Dialogue Generation	JiweiLi, Will Monroe, Alan Ritter, Michel Galley	2016	Subset of 10 million messages from the OpenSubtitles dataset was taken and extract 0.8 million sequences with the lowest likelihood of generating the response.	The learning system consists of two agents. p is used to denote sentences generated from the first agent and q to denote sentences from the second. The two agents take turns talking with each other. A dialogue can be represented as an alternating sequence of sentences generated by the two agents.	Evaluation of dialogue generation systems is done using both human judgments and two automatic metrics: conversation length (number of turns in the entire session) and diversity.	Framework captures the compositional models of the meaning of a dialogue turn and generates semantically appropriate responses.	The dialogue sometimes enters a cycle with length greater than one (repetitive utterances in consecutive turns) The model also sometimes starts a less relevant topic during the conversation.

Serial Number	Title of the paper	Authors	Year of publication	Dataset used	Methodology/ Technology	Performance Metrics	Advantages	Disadvantages
8	Topic Augmented Neutral Response Generation with a Joint Attention Mechanism	Chen Xing, Wei Wu, Jie Liu.	2016	Data sets obtained 5671846 post-response pairs from Sina Weibo3. We then use Standord Chinese word segmenter to tokenize post-response pairs as inputs of all models.	Chose Twitter LDA among various probabilistic topic models because in conversation data, the input posts are short, informal, and contain quite a lot of scattered topics. These characteristics are similar with those of Twitter data and meet the assumptions of Twitter LDA quite well.	Perplexity has been used as an evaluation metric. It measures how well the model predicts a response. A lower perplexity score indicates better generation performance. $PPL = exp\left\{-\frac{1}{N}\Sigma_{i=1}^{N} \log(p(\mathbf{Y}_i))\right\}.$ Human annotations.	TAJA-Seq2Seq and TASeq2Seq decrease the number of fair and plain responses and increase the number of natural and interesting one.	May lead to generic response generation including terminology like "I see","I don't know" which ay severely affect chatbot interaction experience.

Serial Number	Title of the paper	Authors	Year of publication	Dataset used	Methodology/ Technology	Performance Metrics	Advantages	Disadvantages
9	Building Chatbot with Emotions	Honghao WEI, Yiwei Zhao, Junjie Ke	2017	88 million subtitles were collected from an open subtitle database. The subtitles belong to movies and TV shows and are enough to act as appropriate training material for chatbot conversation.	Three layers : First : vanilla seq2seq neural network is used to modify sentiment level correlation. Second : sentiment rewards are introduced during the learning phase. Third : training is done to create an emotional chatbot.	For word coherence and grammatical correctness, perplexity and BLEU are used. Typetoken ratio for unigrams and bigrams in generated response are also used, to avoid generic and boring answers.	A complete well emotional chatbot was introduced, with reasonably positive results for emotional embedding, internal and external conversation memory (both content and emotion level)	Due to time constraints, the training phase of the Deep Reinforcement Learning Model was not completed.
10	Deep Learning Based Chatbot Models	Richard Krisztian Csaky	2017	Cornel Movie Dialogue Corpus, OpenSubtitles Corpus	Tensor Library, Tensor2Tensor training	Qualitative Analysis on the responses of different models, and checking their accuracy	The Seg2Seg model perofms better than the transformer model	The loss function issue, Temporal Conditioning

From the literature survey conducted, some of the main issues regarding chat bots and their utilisations were highlighted. It was from a study of these that the problem statements for this project were derived. As observed, one major issue was the fact that chatterbots were designed to respond to only specific questions, and fell short when the questions were changed to a more conversation like format. This is parallel to the problem of heavy dependance on the training set for nature of response. Another common issue is the inability of the chatterbot to hold a conversation, in the sense, that it does not remember old responses and hence, cannot make informed responses. Yet another problem faced was that after training the model, it seemed to have an affinity towards certain responses. It basically means that the bot was selecting those responses that were frequently used in the language or appeared in the corpus, thus sometimes giving incorrect or loosely fitting responses. For the models that were able to hold a conversation, it was noticed that some bots were mixing up conversations and starting irrelevant discussions. Some generic responses were also recorded for both, conversation and non-conversation type bots. Having studied these primary issues, it was concluded that for this project, the main aim would be to achieve two things: one, build the Chabot in such a way that it is capable of responding to different formats of questions, not only the pre-fed ones in the fixed pre-fed format. Second, the aim was also to reduce the number of generic responses like “I see” or “I don’t know” which meant having a better understanding of keywords and better detection algos for the same. The banking is placed heavily on keywords because it ensures that irrespective of the formation which the user asks the question, the essence of the question is caught and the bot is able to produce a response that is of high accuracy.

3. OVERVIEW OF WORK

3.1 PROBLEM DESCRIPTION

In this day and age of growing technology, a chat bot is the obvious next step. The use of a chat bot is versatile, be it fields of work, education or entertainment. The aim of a Chabot is to provide human-like communication without involving humans, thereby automating large sections of various industries, like customer care etc. The aim of this project is to demonstrate the use of a Chabot on a smaller scale, using 3 subtopics of Hostels, Academics and Extra Curricular, under the main heading of Information Regarding VIT. The aim is to provide a self-developed corpus, reducing the reliance on Python libraries, and to also increase accuracy in communication. In addition to a focus on VIT, this Chabot is also trained in general conversations, cricket, humour, AI, greetings, health, literature, politics, movies, trivia, sports etc.

What is new about the project is that it uses Neural Machine Translation (sequence-2-sequence) model. Sequence-to-sequence (seq2seq) models have enjoyed great success in a variety of tasks such as machine translation, speech recognition, and text summarisation. In earlier projects a Google Neural Machine Translation system was used and it turned out to give good results, but seq2seq are an improvement over GNMT and are predicted to perform better.

3.2 SOFTWARE REQUIREMENTS

Python 3.6 Libraries required:

- Numpy
- Pandas
- Tensorflow-gpu
- Tensorflow
- Nltk
- Sqlite3

4. SYSTEM DESIGN

Our Chabot is specifically designed to respond to all questions related to VIT, and its academics, extracurricular and hostels. We've added an explanation of the various libraries we used to help with the implementation:

Flask

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks. Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy. The Pallets organization develops and supports Flask and the libraries it uses.
For our project, we've used Flask 1.0.2.

Chatterbot

Chatterbot is a machine learning, conversational dialog engine build in Python which makes it possible to generate responses based on collections of known conversations. The language independent design of Chatterbot allows it to be trained to speak any language.
An example of typical input would be something like this:
> **user:** Good morning! How are you doing?
> **bot:** I am doing very well, thank you for asking.
> **user:** You're welcome.
> **bot:** Do you like hats?

An untrained instance of Chatterbot starts off with no knowledge of how to communicate. Each time a user enters a statement, the library saves the text that they entered and the text that the statement was in response to. As Chatterbot receives more input the number of responses that it can reply and the accuracy of each response in relation to the input statement increase. The program selects the closest matching response by searching for the closest matching known statement that matches the input, it then returns the most likely response to that statement based on how frequently each response is issued by the people the bot communicates with.

SQLAlchemy

SQLAlchemy is the Python SQL toolkit and Object Relational Mapper that gives application developers the full power and flexibility of SQL. SQLAlchemy provides a full suite of well-known enterprise-level persistence patterns, designed for efficient and high-performing database access, adapted into a simple and Pythonic domain language.

Major SQLAlchemy features include:

- An industrial strength ORM, built from the core on the identity map, unit of work, and data mapper patterns. These patterns allow transparent persistence of objects using a declarative configuration system. Domain models can be constructed and manipulated naturally, and changes are synchronized with the current transaction automatically.
- A relationally-oriented query system, exposing the full range of SQL's capabilities explicitly, including joins, subqueries, correlation, and most everything else, in terms of the object model. Writing queries with the ORM uses the same techniques of relational composition you use when writing SQL. While you can drop into literal SQL at any time, it's virtually never needed.
- A comprehensive and flexible system of eager loading for related collections and objects. Collections are cached within a session, and can be loaded on individual access, all at once using joins, or by query per collection across the full result set.
- A Core SQL construction system and DBAPI interaction layer. The SQLAlchemy Core is separate from the ORM and is a full database abstraction layer in its own right, and includes an extensible Python-based SQL expression language, schema metadata, connection pooling, type coercion, and custom types.
- All primary and foreign key constraints are assumed to be composite and natural. Surrogate integer primary keys are of course still the norm, but SQLAlchemy never assumes or hardcodes to this model.
- Database introspection and generation. Database schemas can be "reflected" in one step into Python structures representing database metadata; those same structures can then generate CREATE statements right back out - all within the Core, independent of the ORM.

SQLAlchemy’s philosophy:

- SQL databases behave less and less like object collections the more size and performance start to matter; object collections behave less and less like tables and rows the more abstraction starts to matter. SQLAlchemy aims to accommodate both of these principles.
- An ORM doesn’t need to hide the “R”. A relational database provides rich, set-based functionality that should be fully exposed. SQLAlchemy’s ORM provides an open-ended set of patterns that allow a developer to construct a custom mediation layer between a domain model and a relational schema, turning the so-called “object relational impedance” issue into a distant memory.
- The developer, in all cases, makes all decisions regarding the design, structure, and naming conventions of both the object model as well as the relational schema. SQLAlchemy only provides the means to automate the execution of these decisions.
- With SQLAlchemy, there’s no such thing as “the ORM generated a bad query” - you retain full control over the structure of queries, including how joins are organized, how subqueries and correlation is used, what columns are requested. Everything SQLAlchemy does is ultimately the result of a developer- initiated decision.
- Don’t use an ORM if the problem doesn’t need one. SQLAlchemy consists of a Core and separate ORM component. The Core offers a full SQL expression language that allows Pythonic construction of SQL constructs that render directly to SQL strings for a target database, returning result sets that are essentially enhanced DBAPI cursors.
- Transactions should be the norm. With SQLAlchemy’s ORM, nothing goes to permanent storage until commit() is called. SQLAlchemy encourages applications to create a consistent means of delineating the start and end of a series of operations.
- Never render a literal value in a SQL statement. Bound parameters are used to the greatest degree possible, allowing query optimizers to cache query plans effectively and making SQL injection attacks a non-issue.

5. IMPLEMENTATION

a. Description of Modules/Programs

1. The corpuses

```
GNU nano 2.0.6                    File: vit.yml

categories:
- VIT
- Vellore Institute of Technology
conversations:
- - What branches does VIT have?
  - VIT has computer science, electrical, chemical, mechanical and civil enginering. It also has a biotech, and architecture department.
- - What does SAS stand for?
  - School of Advanced Science.
- - What does V SPARC stand for?
  - School of Architecture.
- - What does SBST stand for?
  - School of Bio Sciences and Technology.
- - What does SELECT stand for?
  - School of Electrical Engineering.
- - What does SCOPE stand for?
  - School of Computer Science and Engineering
- - What does SENSE stand for?
  - School of Electronics and Engineering.
- - What does SMEC stand for?
  - School of Mechanical Engineering.
- - What does SITE stand for?
  - School of Information Technology and Engineering.
- - What does SSL stand for?
  - School of Social Science and Languages.
- - What does FFCS stand for?
  - Fully Felxible Credit System
- - Do the students have their own login?
  - Yes, each student has their own ID and password, that mantains all of their information including their hostel details, and marks. It's also used for accessing their ci$
- - Do the parents have access to their students marks?
  - Yes, there is also a parent login through which they can access their students marks and stay updated.
- - Does VIT have any academic centres?
  - Yes, it has the Academic Staff College, the VIT Technology Business Incubator, GMAT Center, and the Survery Research Center.
- - Does VIT have any clubs or chapters?
  - Yes!
- - How many clubs and chapters are there in VIT?
  - There are over 120 clubs and chapters at VIT.
- - Can we join clubs/chapters in our first year?
  - Yes, but not in the first semester.
- - What all literary clubs are there at VIT?
  - Debate Society, MUN Society, English Literary Association, Hindi Literary Association.
- - What all NGOs are there at VIT?
  - There's JC, BIF, GGB, AIESEC, Anokha.
- - What all technical chapters are there at VIT?
  - IEEE, SEDS, IEEE-CS, ACM, AICHE, IICHE
```

```
GNU nano 2.0.6                    File: vit.yml

- - Can students be a part of more than one club or chapter?
  - Yes.
- - How can students learn more about these clubs and chapters?
  - Students can check the VIT page or attend the annual club and chapter expo to know more.
- - Do clubs and chapters help in personal development of students?
  - Yes.
- - How many events take place in VIT?
  - Multiple events happen in VIT on a daily basis.
- - How many hostels are there in total for the students at VIT?
  -  23 hostels.
- - How are the hostels divided?
  - 17 men's hostel and 6 women's hostel.
- - How many students can be accommodated in the hostels?
  - 13000 male students and 4300 female students.
- - What additional facilities are made available?
  - Gym and swimming facilities are available for both male and female students.
- - What is the division of rooms for male students?
  - Single, double, three, four and six bedded rooms.
- - What is the division of rooms for female students?
  - Single, double, four and six bedded rooms.
- - Are all rooms bunker or do Individual bedded rooms exist?
  - Not all rooms have bunker beds, some rooms are equipped with separate beds.
- - What is the status of air conditioning services in the rooms?
  - Both AC and non-AC rooms are available for students.
- - How does room allocation take place?
  - On merit basis, based on the students' NCGPA.
- - How do students select their roommates?
  - Of their own will, the room taker carries all IDs to the counselling hall.
- - How are the rooms furnished?
  - The rooms have appropriate number of cupboards, study tables, chairs, fans, tube lights, mirrors, windows and cabinets.
- - Can the students decorate their rooms?
  - As long as the decorations do not harm the rooms, yes.
- - What are the eating facilities available?
  - There are shops available in and around hostels for the students to buy food.
- - What is the warden system like?
  - Each block has a couple of wardens, who take shifts in rotation. There are Associate Chief Wardens, Chief Wardens and Deputy Chief Wardens too.
- - What is the process for room switching?
  - The students get to change rooms every year. Their belongings are kept in the dorms inside the hostels for the vacations.
- - What is the mess system like?
  - There are multiple messes, by different caterers.
- - Who all are allowed in the rooms?
  - Only students and college officials (wardens, cleaners) are allowed in the rooms. Parents, relatives, siblings etc are not allowed.
- - What fun activities are arranged for students in hostels?
  - Occasional movie nights are held. Festivals are celebrated in open areas like the basketball court in the girls hostel and the outdoor stadium in the boys hostel.
```

2. The Interface

A. index.html

```html
<!DOCTYPE html>
<html>
```

```
  <head>
    <link rel="stylesheet" type="text/css" href="/static/style.css">
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
  </head>
  <body>
    <h1>VITTY</h1>
    <h3>VIT'S Chatbot</h3>
    <div>
      <div id="chatbox">
        <p class="botText"><span>Hi! I'm Vitty.</span></p>
      </div>
      <div id="userInput">
        <input id="textInput" type="text" name="msg" placeholder="Message">
        <input id="buttonInput" type="submit" value="Send">
      </div>
      <script>
        function getBotResponse() {
          var rawText = $("#textInput").val();
          var userHtml = '<p class="userText"><span>' + rawText + '</span></p>';
          $("#textInput").val("");
          $("#chatbox").append(userHtml);
          document.getElementById('userInput').scrollIntoView({block: 'start',
behavior: 'smooth'});
          $.get("/get", { msg: rawText }).done(function(data) {
            var botHtml = '<p class="botText"><span>' + data + '</span></p>';
            $("#chatbox").append(botHtml);
            document.getElementById('userInput').scrollIntoView({block: 'start',
behavior: 'smooth'});
          });
        }
        $("#textInput").keypress(function(e) {
            if(e.which == 13) {
                getBotResponse();
            }
        });
        $("#buttonInput").click(function() {
          getBotResponse();
        })
      </script>
    </div>
  </body>
</html>
```

B. style.css

```
body {
    font-family:sans-serif;
}

h1 {
    color: black;
    margin-bottom: 0;
    margin-top: 0;
    text-align: center;
    font-size: 40px;
}

h3 {
    color: black;
    font-size: 20px;
    margin-top: 3px;
    text-align: center;
}
```

```
#chatbox {
    margin-left: auto;
    margin-right: auto;
    width: 40%;
    margin-top: 60px;
}

#userInput {
    margin-left: auto;
    margin-right: auto;
    width: 40%;
    margin-top: 60px;
}

#textInput {
    width: 87%;
    border: none;
    border-bottom: 3px solid #009688;
    font-family:sans-serif;
    font-size: 17px;
}

#buttonInput {
    padding: 3px;
    font-family:sans-serif;
    font-size: 17px;
}

.userText {
    color: white;
    font-family:sans-serif;
    font-size: 17px;
    text-align: right;
    line-height: 30px;
}

.userText span {
    background-color: #009688;
    padding: 10px;
    border-radius: 2px;
}

.botText {
    color: white;
    font-family:sans-serif;
    font-size: 17px;
    text-align: left;
    line-height: 30px;
}

.botText span {
    background-color: #EF5350;
    padding: 10px;
    border-radius: 2px;
}

#tidbit {
    position:absolute;
```

```
        bottom:0;
        right:0;
        width: 300px;
    }
```

b. Source Code

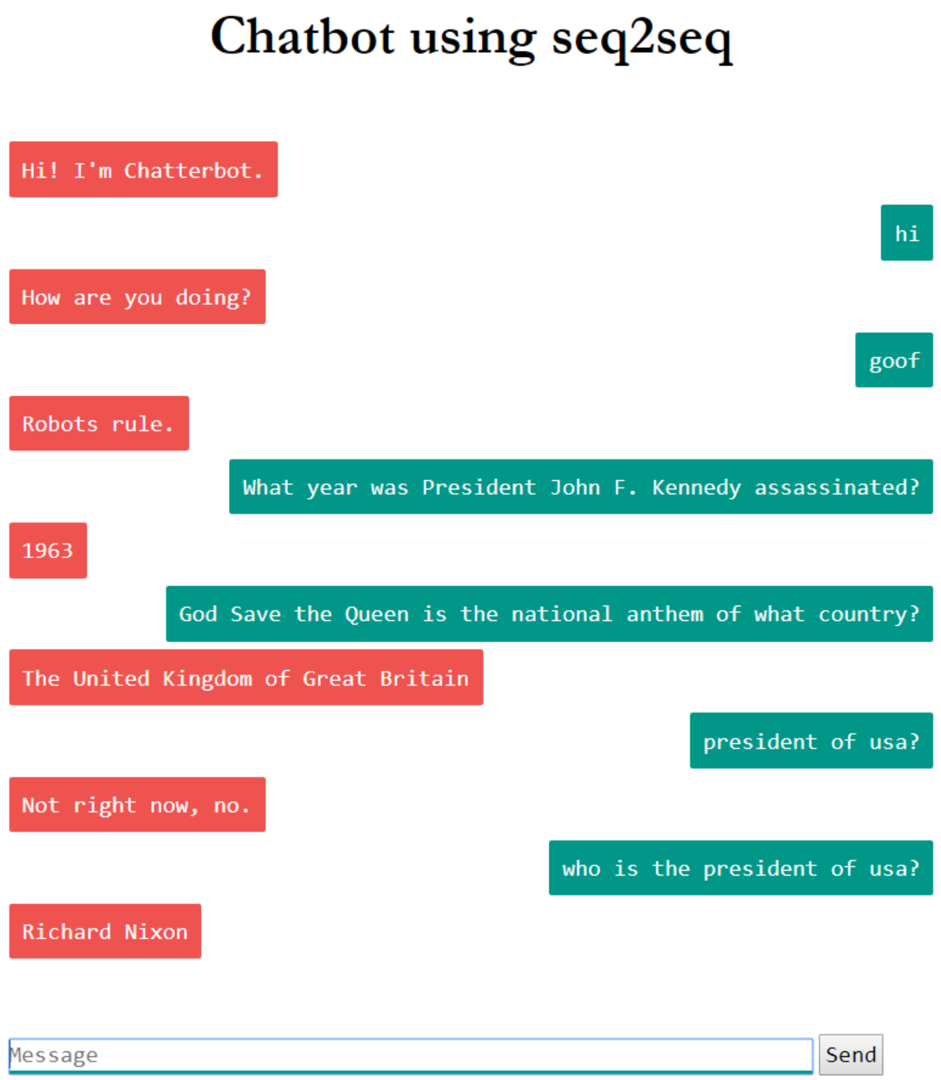
```
from flask import Flask, render_template, request
from chatterbot import ChatBot
from chatterbot.trainers import ChatterBotCorpusTrainer

app = Flask(__name__)

english_bot = ChatBot("Chatterbot",
storage_adapter="chatterbot.storage.SQLStorageAdapter")
trainer = ChatterBotCorpusTrainer(english_bot)
trainer.train("chatterbot.corpus.english")


@app.route("/")
def home():
    return render_template("index.html")

@app.route("/get")
def get_bot_response():
    userText = request.args.get('msg')
    return str(english_bot.get_response(userText))


if __name__ == "__main__":
    app.run()
```

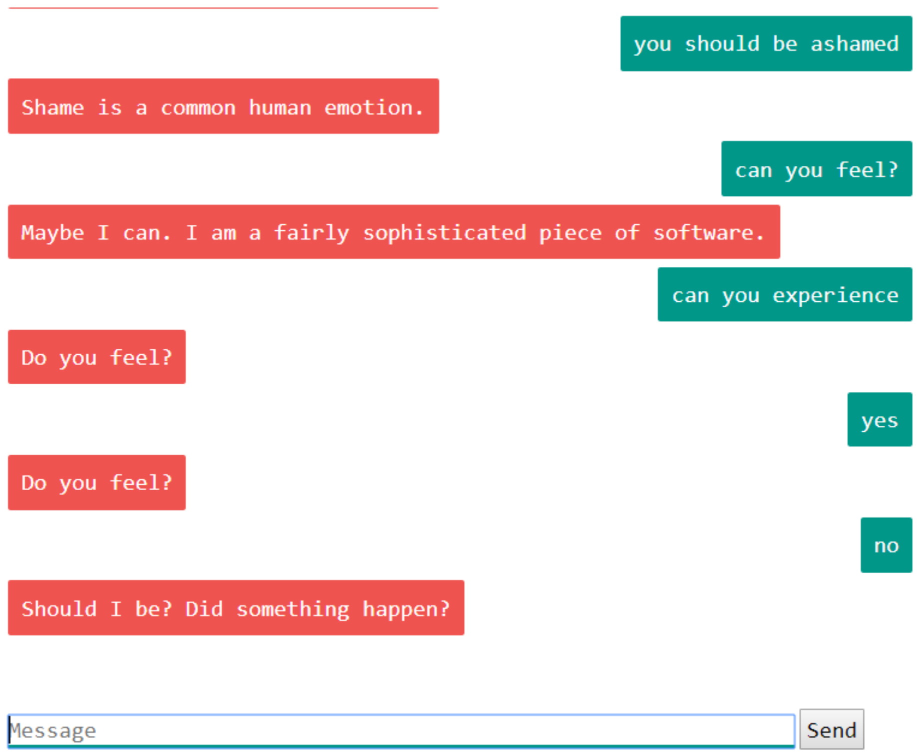
c. Test cases (for the VIT corpus)

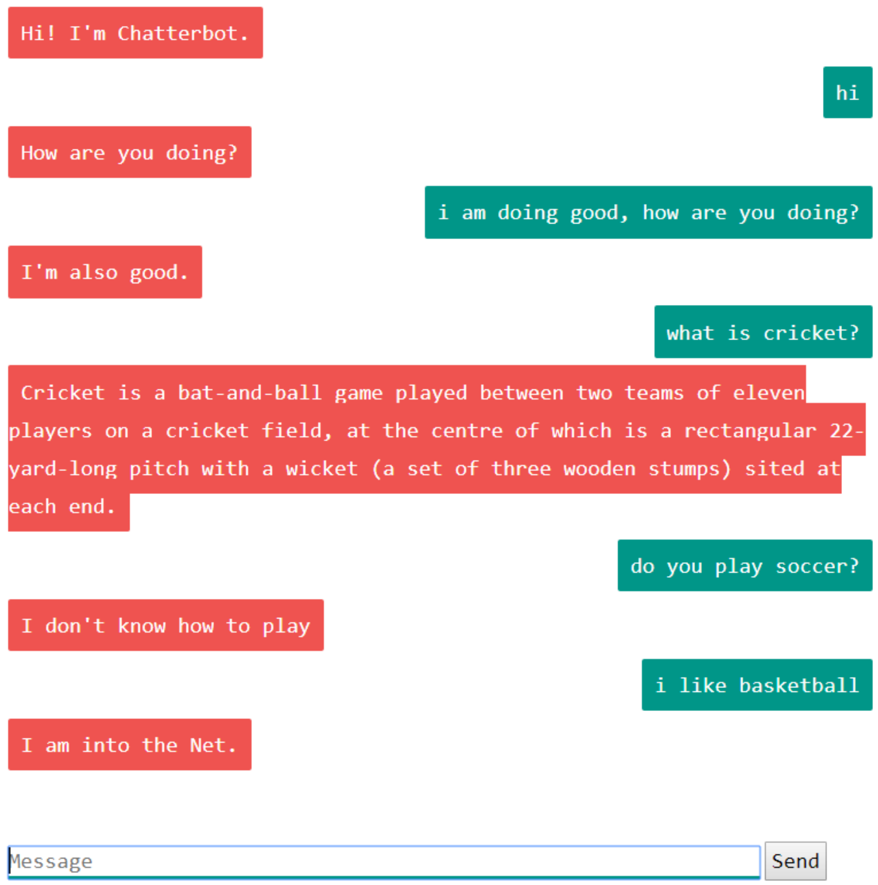
S. No	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1.	Check bot's response to questions on academics	What does SCOPE stand for?	School of Computer Science Engineering	School of Computer Science Engineering	**Pass**
2.	Check bot's response to questions on hostels	Are all rooms bunkers or do Individual	Not all rooms have bunker beds, some rooms are	Not all rooms have bunker beds, some rooms are	**Pass**

		bedded rooms exist?	equipped with separate beds.	equipped with separate beds.	
3.	Check bot's response to questions on extra circulars	What all NGOs are there at VIT?	There's JC, BIF, GGB, AIESEC, Anokha	There's JC, BIF, GGB, AIESEC, Anokha	**Pass**
4.	Check bot's response to numerical responses	How many hostels are there in total for the students at VIT?	23 hostels.	23 hostels.	**Pass**

d. Execution snapshots


Chatbot using seq2seq
Hi! I'm Chatterbot.
hi
How are you doing?
goof
Robots rule.
What year was President John F. Kennedy assassinated?
1963
God Save the Queen is the national anthem of what country?
The United Kingdom of Great Britain
president of usa?
Not right now, no.
who is the president of usa?
Richard Nixon
Message
Send


you should be ashamed
Shame is a common human emotion.
can you feel?
Maybe I can. I am a fairly sophisticated piece of software.
can you experience
Do you feel?
yes
Do you feel?
no
Should I be? Did something happen?
Message
Send


Hi! I'm Chatterbot.
hi
How are you doing?
i am doing good, how are you doing?
I'm also good.
what is cricket?
Cricket is a bat-and-ball game played between two teams of eleven players on a cricket field, at the centre of which is a rectangular 22-yard-long pitch with a wicket (a set of three wooden stumps) sited at each end.
do you play soccer?
I don't know how to play
i like basketball
I am into the Net.
Message
Send

6. CONCLUSION AND FUTURE DIRECTIONS

Via this project the purpose of a Chabot has been displayed on a small scale. It is safe to conclude that Chatbots are a very human friendly and fast way to conduct communication in spheres where the presence of a human is not necessarily needed. These could include customer care, direction systems, or simple companion apps. This can allow organisations to direct their human resources to more human centric job requirements. The Chabot that was created was successful in answering questions regarding hostels, academics and extracurricular activities that are available in VIT. For future instances the corpus can be expanded to answer a variety of other questions about VIT, and then beyond VIT as well. The corpus may cover any field and answer any number of questions with slight modifications.

7. REFERENCES

[1] Zheng Xu, Sheng Wang, Feiyun Zhu, Junzhou Huang, “Seq2seq Fingerprint: An Unsupervised Deep Molecular Embedding for Drug Discovery”, ACM-BCB '17 Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, 2017, ISBN: 978-1-4503-4722-8, Pages 285-294.

[2] WenqiHou,YunNie,“Seq2seq-AttentionQuestionAnsweringModel”, Standford University.

[3] ZiYin,Keng-haoChang,RuofeiZhang,“DeepProbe:InformationDirected Sequence Understanding and Chatbot Design via Recurrent Neural Networks”, KDD '17 Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, Pages 2131-2139.

[4] PingleiGuo,YusiXiang,YunzhengZhang,WeitingZhan,“Snowbot:An empirical study of building chatbot using seq2seq model with different machine learning framework”, Semantic Scholar.

[5] DeepLearningforChatbots,Part1–Introduction.

[6] Mart í n Abadi et al. “Tensorflow: Large-scale machine learning on heterogeneous distributed systems”. In: arXiv preprint arXiv:1603.04467 (2016).

[7] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. “Sequence to sequence learning with neural networks”, Advances in neural information processing systems, 2014, pp. 3104–3112.

[8] Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, Zhi Jin, “Sequence to Backward and Forward Sequences: A Content-Introducing Approach to Generative Short-Text Conversation”, 2016, arXiv:1607.00970 [cs.CL].

[9] Generative Indonesian Conversation Model using Recurrent Neural Network with Attention Mechanism by Andry Chowanda, Alan Darmasaputra Chowanda (2018)

[10] Implementation of A Neural Natural Language Understanding Component for Arabic Dialogue Systems Component for Arabic Dialogue Systems by Abdallah M. Bashir, Abubakr Hassan, Benjamin Rosman , Daniel Duma, ,Mohanad (2018)