PROJECT REPORT

ON

IMPLEMENTATION OF AI CHATBOT BASED ON ROUTES IN DHAKA CITY

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

JULY 2020

IMPLEMENTATION OF AI CHATBOT BASED ON ROUTES IN DHAKA CITY

BY

MD. ZAHID HASAN
ID: 162-15-775
AND

MD. YEASHIN RAHMAN
ID: 162-15-750
AND

MD. ABDUR ROUF NIROB
ID: 162-15-741

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

MD. NAZMUL HOQ
Lecturer & Associate Head
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

JULY 2020

APPROVAL

This Project titled "**Implementation of AI chatbot based on routes in Dhaka city**", submitted by Md. Zahid Hasan (ID 162-15-775) and Md. Yeashin Rahman (ID 162-15-750) and Md. Abdur Rouf Nirob (ID 162-15-741) to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation was held on Sunday, 12 July, 2020.

BOARD OF EXAMINERS

Dr. Syed Akhter Hossain **Chairman**
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Mr. Narayan Ranjan Chakraborty **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Mr. Md. Nazmul Hoq **Internal Examiner**
Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Mohammad Shorif Uddin **External Examiner**
Professor and Chairman
Department of Computer Science and Engineering
Jahangirnagar University

DECLARATION

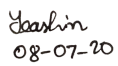
We hereby declare that this project has been done by us under the supervision of **Md. Nazmul Hoq, Lecturer & Associate Head, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

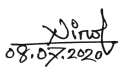
Md. Nazmul Hoq
Lecturer & Associate Head
Department of CSE
Daffodil International University

Submitted by:

Md. Zahid Hasan
ID: 162-15-775
Department of CSE
Daffodil International University



Md. Yeashin Rahman
ID: 162-15-750
Department of CSE
Daffodil International University



Md. Abdur Rouf Nirob
ID: 162-15-741
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty Allah for His divine blessing makes us possible to complete the final year project successfully.

We are really grateful and wish our profound indebtedness to **Md. Nazmul Hoq**, Lecturer & Associate Head, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "*Artificial Intelligence*" to carry out this project. His endless patience ,scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to **Prof. Dr. Syed Akhter Hossain**, Head**,** Department of CSE, for his kind help to finish our project and also to other faculty members and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patience of our parents

ABSTRACT

With the development of Machine Learning & Artificial Intelligence, chatbots are being used in many areas. The problem we are trying to solve here is which bus circles around on which route. A friendly chatbot is used to answer the questions automatically. Qodra chatbot is a fully data driven retrieval based closed domain chatbot which can respond to the user based on the knowledge and interaction with the users. In our proposed system, users will ask Qodra chatbot questions about going from one place (source place) to another place (destination place) through web & mobile app interaction in natural language. Qodra will collect the statement and search it on it's knowledge database for a similar answer. Based on the different logic adapter it will choose a response with the highest confidence value. Qodra will learn & store from the users' conversations and the dataset must be set & prepared before deployment. Statistically Qodra is able to produce 70-75% accurate response.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Background and History of Chatbot

A chatbot is a piece of software that conducts a conversation via auditory or textual methods. Most people are interested in the programs that are human-like and that is a great opportunity to deal with users as long as the subject is straightforward and falls into the predictable capabilities. The development of Chatbot started around the 19th century. First Chatbot ELIZA was built in 1966 and later in PARRY 1972. ELIZA was able to fool the users for some moment that they were conversing with real human.These types of Chatbot used to require exclusively typed conversation.

Modern Chatbots are smart and can communicate in natural languages. Since it is a program, everybody will have the chance to train their own bot. In the support area there are always required people who need to respond continuously to make a better interaction. In some areas human resources are expensive and limited. Chatbots can help to solve these types of problems.

1.2 Purpose of building Qodra Chatbot

The problem we are trying to solve here is which bus circles around on which route in Dhaka City. Qodra chatbot is a goal oriented closed domain chatbot. A friendly bot is used to answer the questions automatically. Since Chatbots are always available for service, users don't have to find the bus names manually. This Chatbot will be able to respond both in English and Bangla. Here, users will ask Qodra to tell about the available bus from one place to another, the Chatbot will try to understand the language and will generate a response based on the input statement. Qodra is also capable of making some

interesting conversation. DU MAMABOT, GOLPO Chatbots are the main inspiration to build Qodra Chatbot.

1.3 Overview Structure

This project report includes seven chapters. Chapter one introduces some history about chatbot and the purpose of building Qodra chatbot. Chapter two presents some basic ideas about machine learning and NLP. Chapter three includes the tools and technologies used to build the Qodra Chatbot. Chapter four describes how Qodra chatbot is fed with corpus data and how these dataset is collected and generated to obtain the goal. Chapter five describes how the Qodra chatbot is responding and Testing the responses. Chapter six presents how Qodra Chatbot is integrated in the django web application. Chapter seven is about improving the responses, limitations and future plans.

CHAPTER 2
TECHNICAL ANALYSIS

2.1 Basics of Machine Learning

Machine Learning is one of the search fields of the Artificial Intelligence (AI) area. Machine learning is a way where we can solve our daily problem by using AI. It begins with reasoning, then knowledge and then learning.

The main idea of Machine Learning is that everything should be measured Quantitatively. The main foundation of Machine Learning is Data. Programmers assign different types of algorithms to train a model . After the training the module which is trained will have knowledge about the data and will produce response.

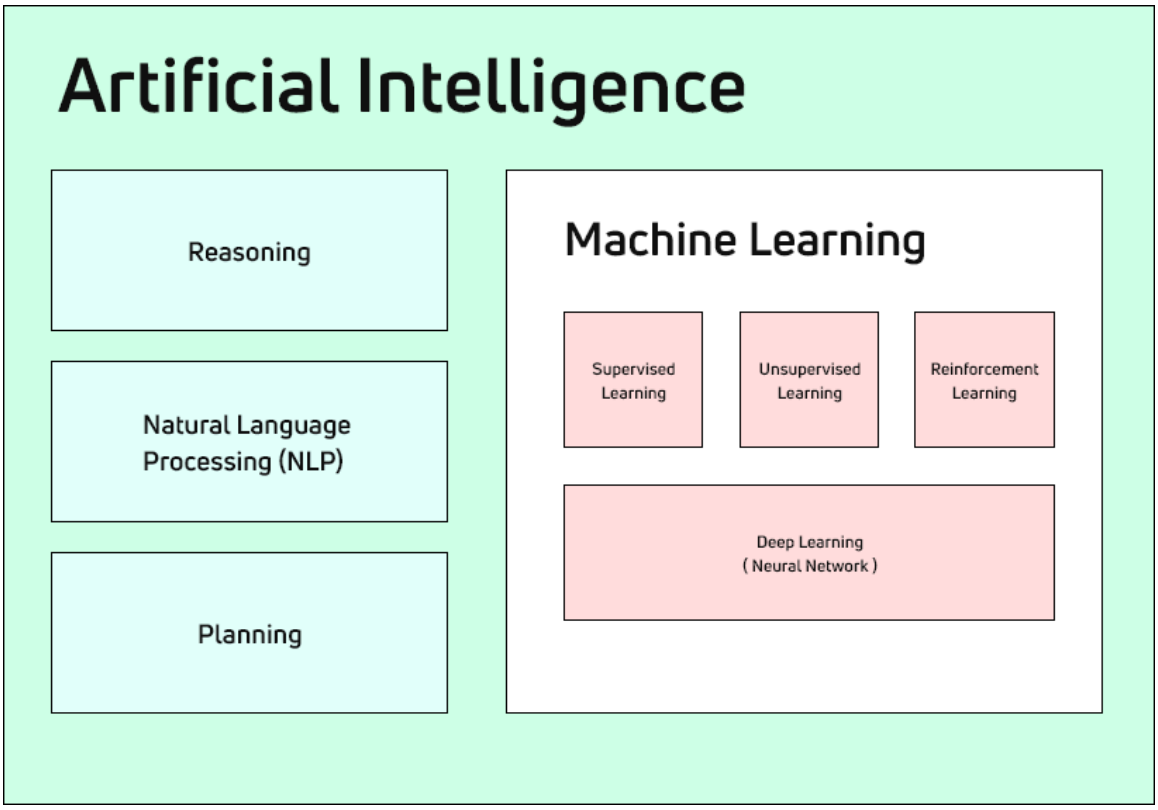


Figure 2.1.1 : Machine Learning Research Area

2.2 Basics of Natural Language Processing

Natural Language Processing is based on two main concepts. First one is Natural Language and the second one is Processing. By Natural Language we mean the way humans express their thinking, the way they speak, write and exchange information. All the languages belong to the natural language like English, Chinese, Bangla, Hindi, Russian etc. But processing is a different matter. Machines cannot process text information like humans do. It has its own way of processing text data. Now , the Natural Language Processing means users will input the program in natural language as human speaks , then the machine will take the input and will run through algorithms and other processing modules and generate a response. Then it'll convert the response into natural language and return to the user console.

NLP are used in many areas such as question and answering, conversation, information extracting, object detection via conversation to the bot etc. Google , Amazon , Microsoft these giant tech companies are building the most powerful bot to manipulate different tasks automatically by voice recognition.

CHAPTER 3
TOOLS AND TECHNOLOGIES

3.1 Python

Python is an interpreted, high-level, general-purpose programming language []. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace.

With the development of AI python is advancing their packages, libraries and frameworks. However there is a significant difference between Python 2 & Python 3. To build the Qodra Chatbot we used python 3.7 version. At least python 3.6 is preferrable.

3.2 Tensorflow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks [1]. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is mainly used in Google for Research and Development purposes.

History:

Around 2011, Google Brain developed a Machine Learning system named DistBelief based on neural networks which was used in Google's parent company known as Alphabet. It was used for both research and commercial purposes. TensorFlow is Google Brain's 2nd generation system. It can run on multiple GPUs & CPUs. TensorFlow is also available for linux, windows & Mac.

Advantage:

Training Model is adaptable and simple to learn. It has a high level API, for example, Keras and Slim which is very powerful for planning in AI calculations. Furthermore, it has one of the most powerful AI libraries in the world, it has numerous applications and APIs, which can easily be used in further development.

Brief Introduction:

What is Tensor ? A mathematical Tensor is described as a K-dimensional vector. A K-dimensional dataset can be represented by a single tensor. See figure below

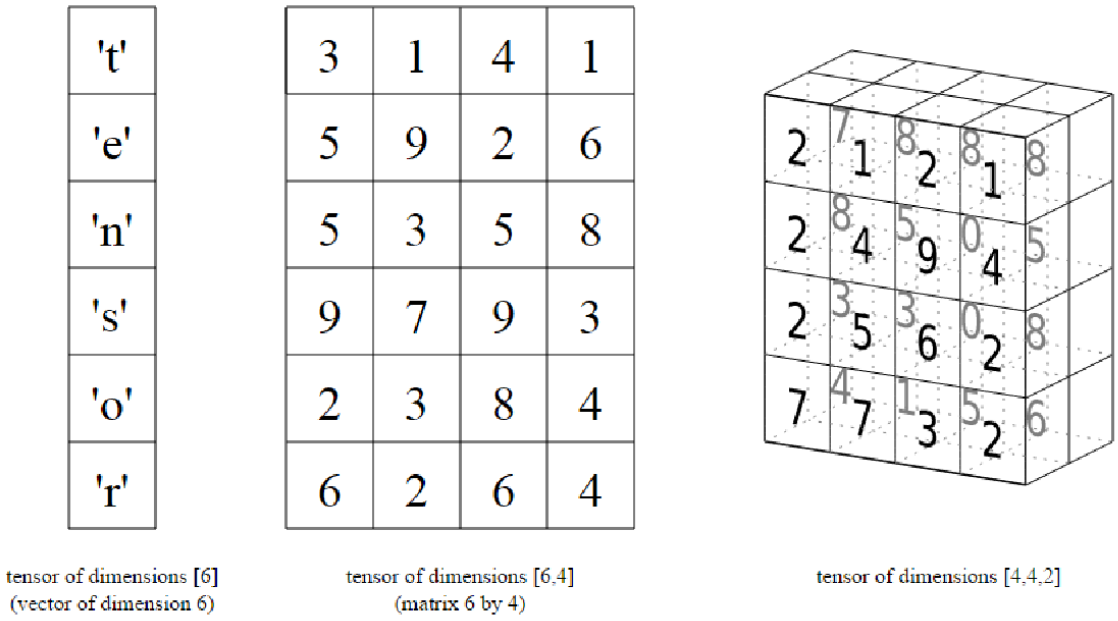


Figure 3.2.1 : Tensor

Data will be more complex when the dimension is increased. In extreme complex dimension data is hard to understand and present.

What is Flow? Flow is like a tree structure in that case. Each node in tree structure represents a calculation operation of any equation. After finishing every operation a new tensor will be produced.

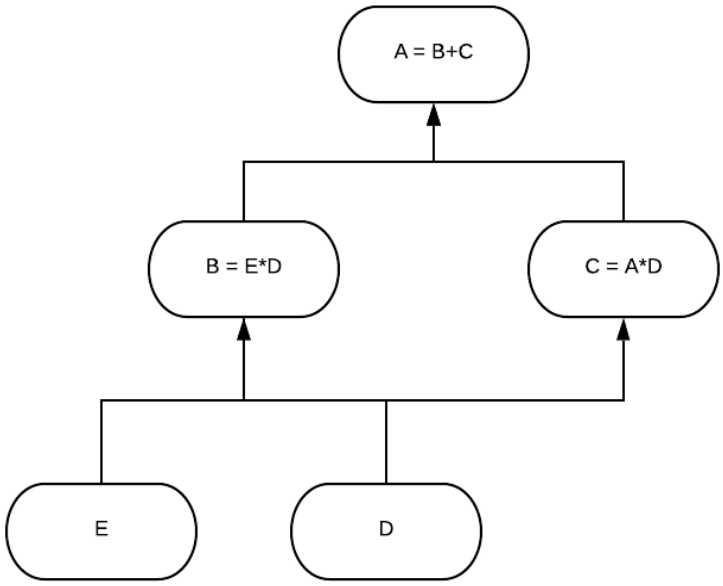


Figure 3.2.2 : Flow

At the Figure # shows the progress of function A = B + C. Here the lowest node E & D is always a tensor. In that case it has dependency that cannot start calculation from the beginning of the chart. That's why in every calculation it will produce a new tensor and child nodes will work as input of the calculation. Each operation is independent here , and will complete each operation simultaneously.

That’s where Tensorflow shows it’s beauty. All the tensor runs automatically and parallely and completes its tasks. For the higher dimensions of the tensor higher machine power is required.

3.3 NLTK

Natural Language Toolkit is also known as NLTK [2]. NLTK is a free, open source & community- driven project. It is a leading platform for building powerful python programs to deal with natural human language data. It has around 50 Corpora data and lexical resources such as WordNet, along with some powerful text processing packages for classification, tokenization, tagging, parsing, stemming and semantic reasoning python wrappers for production level NLP libraries.

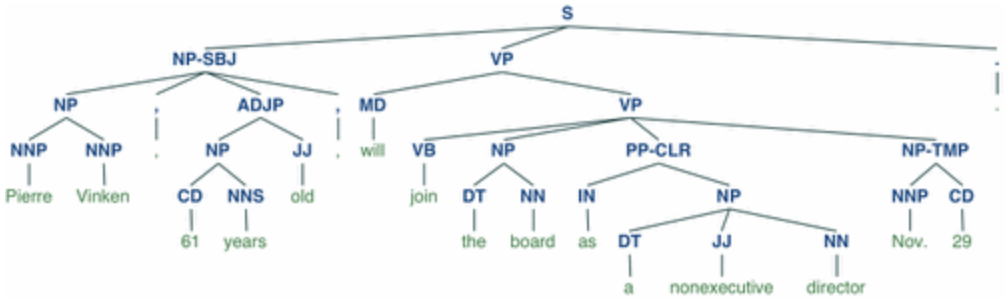


Figure 3.3.1 : NLTK

3.4 WordNet Interface

WordNet is just another NLTK corpus reader. WordNet contains the relationship between two words. It has different kind of structure , the most commonly used structure is “Sysnet”. If we can think of WordNet as a database then Systnet is like a primary key. It can be used to get the meaning of words , usage examples and definition. Lemmas are known to be the collection of similar words. It is a graph structure where nodes represent words itself and edges represent relationships between the words. This is not only the relationship between the synonyms collection but also associates with certain relationships such as upper and lower , whole & partial relationship, antisense relationship, inheritance relationship etc.

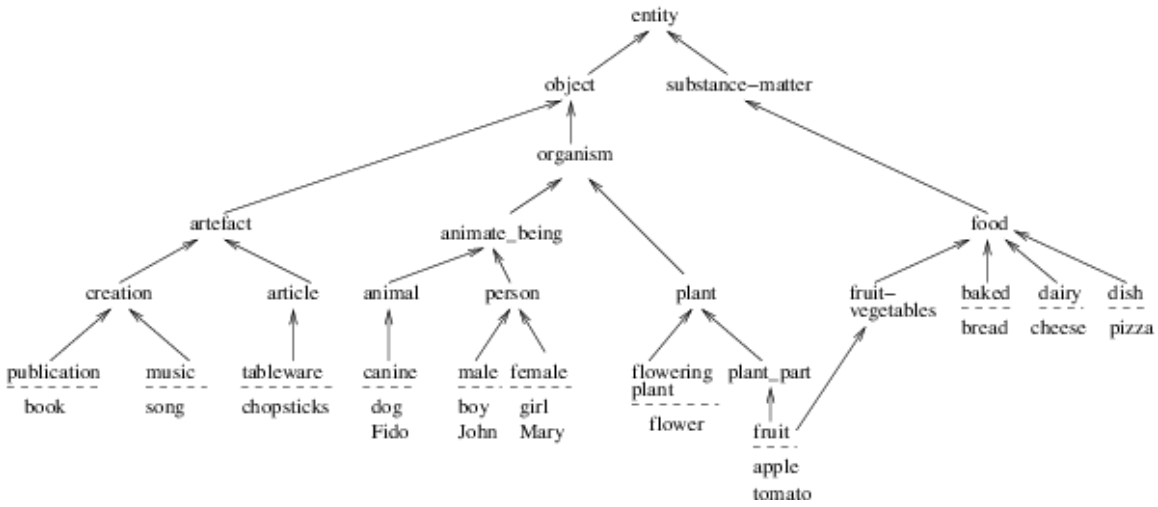


Figure 3.4.1 : WordNet Interface [3]

3.5 Chatterbot

Overview:

Chatterbot is a powerful python library, which can easily generate responses based on a user's statement. A chatter engine can be easily built by Chatterbot library. It is combined with different types of Machine Learning algorithms to make different types of response. In that project we used chatterbot version 1.0.4.

Language Support:

Chatterbot is combined with the NLTK . NLTK has more than 50 corpora data in different languages. With that capability Chatterbot becames language independent. Following that chatterbot allows itself to train any language data. It is specifically designed to learn from Chat data. Besides, chatterbot allows us to learn from the conversation.

Principal:

At the very beginning chatterbot instance has no database and it has no knowledge about anything. It also doesn't know how it will be able to communicate with the user. Once a user starts to converse with the bot , it will save the input statement and responses to it's library model. When the user will start to coversing more and more the bot instance will be able to produce higher accurate results. Based on the input statement , bot tries to find the best match result to the response.

Training:

Chatterbot comes with different types of handy features. Since every chatbot needs to be designed in their specific way. So it needs to be trained with the specific data to generate the correct response. Chatterbot has a small module to train the bot with the corpus data. When the training process begins it starts to load some sample data to the program database. It will create some graph data structure that shows the user the statement and known responses. After putting heavy dataset chatterbot puts these dataset into the knowledge structure to make the accurate response. These python tools made it very easy to develop chatbot.

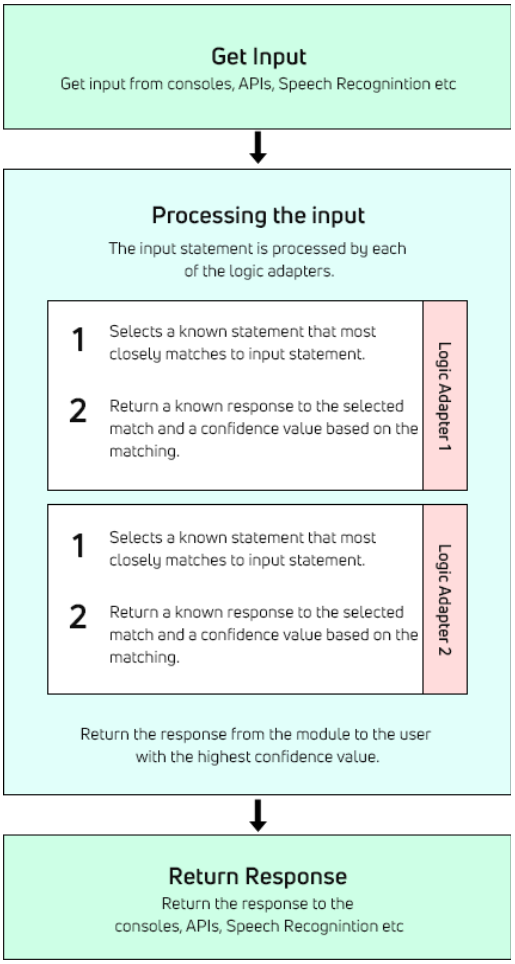


Figure 3.5.1 : Chatterbot process flowchart

Training Preparation:

Training the corpus data is one most important part of building chatbot. It is built in with different sets of corpus data. It is also included with Bangla corpus data to make some basic conversation.

all the corpus data will be saved on the local machine database. It is also possible to save this data onto the server machine database.

Training Process Diagram :

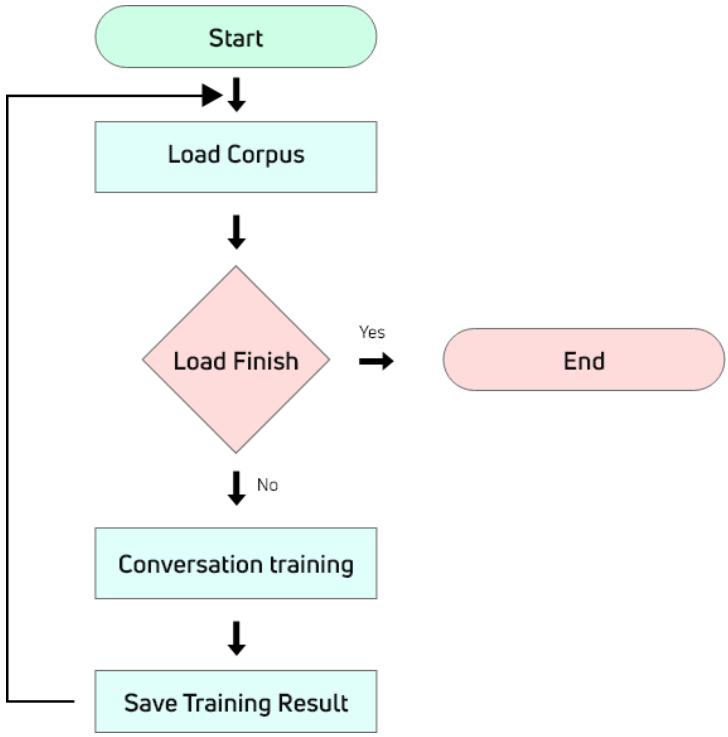


Figure 3.5.2 : Chatterbot training process

Some tasks should be done before starting to train the bot. All corpora data should be in the local machine alongside the python file in the specific directory. It must be loaded into memory or into the machine where the python file will be executed. In case of training with conversation data , the machine tries to understand it from the conversation and store it to the knowledge database. Model will be trained for once then chatterbot will store the training result into the directed file or any specific configured database.

Comparisons:

Chatterbot uses statement objects to hold the results. Another important part is how Chatterbot is selecting the response based on the ability to compare two objects. There are a variety of ways to do this task. Chatterbot comes with some built-in modules to select the response.

Advantage:

- The training corpus or result should be saved on many types of media.
- It supports different types of algorithms such as similarity matching, mathematical estimation algorithm etc.
- A chatbot can be trained in many languages.

Disadvantage:

- Larger corpus data requires long time to be trained.
- It will also take long time to respond.

3.6 Google Colab

Google Colab is a python notebook that comes with a huze library. It also provides powerful CPUs, GPUs, and a large amount of memory space. Users can train their own data using free GPU. It also supports TensorFlow, BigQuery, Google Drive, Mathplotlib, Numpy , Pandas and other basic libraries of Deep Learning. Most interesting part is that we can also add our personal or necessary libraries. The disadvantage here is that saving libraries will vanish if the session is broken. Colab supports python 2 and also python 3 which is user friendly.

3.7 C++

C++ is a general-purpose programming language created by Bjarne Stroustrup as an extension of the C programming language, or "C with Classes". To build the Qodra bus route corpus dataset we have built a module to read and write the dataset in a faster way.

3.8 Virtual Environment - venv

Python supports a package called venv. venv stands for virtual environment. venv allows users to create isolated projects in the same directory.Virtual environment does not interfere with other libraries of other environments. Venv creates a fully lightweight isolated environment for each project. Each environment has its own python library files. Also it is possible to work with the matching python version. It is very easy to use the virtual environments and control it. Only a few lines of command can activate the virtual environment. Once it is activated it is a different environment & also doesn't depend on Operating System Libraries. Python code runs from that environment.

CHAPTER 4
TECHNICAL SPECIFICATION

4.1 Data Preparation

4.1.1 Data Gathering

Data is the base foundation of Artificial Intelligence and other related fields. Since our chatbot is goal based on the problem, we had collected some portion of the dataset manually and with the help of that we generated the rest of the dataset. We followed different types of patterns to obtain the best result. Qodra chatbot learns from chat data and other corpus data. Since we had to generate a dataset in a pattern we wrote a simple program to generate these datasets. So we structured the dataset in corpus dataset manner.

At first we collected the names of the bus stoppages in Dhaka city starting from abdullahpur. It ends in Narayanganj. We have registered around ~ 130 bus stopage names. Then our target was to collect the bus names available on these routes. Then we collected and stored around ~130 bus names. Then our target was to generate the dataset which bus routes on which path. We have built a C++ program which will take the data set of bus list and bus stoppage names then it will generate the dataset based on the available buses on the route. Some examples are given how these datasets are formed.

Table 4.1.1.1 : Basic Structure of Data Preparing

Stopages Names	BUS Names	Buses in bus stop
300 feet	13 No	*Dhour Chourasta*
Aarong	6 No	==
Abdullahpur	7 No	Bkash
Adabor	8 No	Alif
Adamjee College	Achim Paribahan	Binimoy
Agargaon	Agradut-Boishakhi	Manjil
Airport	Akash	Bhuiyan
Amin Bazar	Akik Paribahan	Victor Classic
Ansar Camp	Al Madina Plus One	Ashulia Classic
Arambagh	Al Makka Transport	!=
Asad Gate	Alif	*Kalu Miah Haji Market*
Ashulia	Alike	==
Ashulia Bazar	Anabil	Bkash
Azimpur	Ark Transport	Alif
Azompur	Ashirbad Paribahan	Binimoy
Badda	Ashulia Classic	Manjil
Bakshi Bazar	Asmani Paribahan	Bhuiyan
Banani	Ayat	Victor Classic
Bangla College	Azmeri Glory	Ashulia Classic
Bangla Motor	BRTC	!=
Bashtola	Bahon	Prottasha
Bashundhara	Dewan	==
Bata Signal	Dhaka Chaka	Bkash

At first we designed the dataset in bus stopage wise, Which buses stop on which stoppage. Then we converted the data into the conversational data. After that we build corpus data to feed the machine.

4.1.2 Structuring as Corpus Data

We are using a chatterbot engine to handle the conversational data. At first the bot has no knowledge about anything. It needs to be trained with a specific dataset. Chatterbot engines have some built-in corpus dataset to gain knowledge about the outside world. We have tried to implement our prepared dataset into corpus data. Some Python , C++ programs are written to obtain that objective. Since the Qodra bot will learn from conversational data, we generated the dataset in questions and answer patterns. We tried different types of patterns in the dataset. At first we tried to generate multiple bus names in a single question. But the response was poor. In that case file size was around 35 Mb. Then we built an efficient way of building corpus data where dataset file size reduced to 3 MB and response was much better and faster. Some sample datas is provided below.

Table 4.1.2.1 : Basic conversational training data

Questions	Answers
Which Bus i should take to go from 300 feet to Aarong	Bus available on this route: BRTC
Which Bus i should take to go from 300 feet to Abdullahpur	Bus available on this route: BRTC Sorry, No bus on this route
Which Bus i should take to go from 300 feet to Adabor	Sorry, No bus on this route
Which Bus i should take to go from 300 feet to Adamjee College	Sorry, No bus on this route Bus available on this route: BRTC
Which Bus i should take to go from 300 feet to Agargaon	Bus available on this route: BRTC
Which Bus i should take to go from 300 feet to Airport	Bus available on this route: BRTC
Which Bus i should take to go from 300 feet to Amin Bazar	Bus available on this route: BRTC
Which Bus i should take to go from 300 feet to Ansar Camp	Bus available on this route: BRTC Sorry, No bus on this route
Which Bus i should take to go from 300 feet to Arambagh	Sorry, No bus on this route
Which Bus i should take to go from 300 feet to Asad Gate	Sorry, No bus on this route
Which Bus i should take to go from 300 feet to Ashulia	Bus available on this route: BRTC
Which Bus i should take to go from 300 feet to Ashulia Bazar	Bus available on this route: BRTC Sorry, No bus on this route
Which Bus i should take to go from 300 feet to Azimpur	Bus available on this route: BRTC
Which Bus i should take to go from 300 feet to Azompur	Bus available on this route: BRTC
Which Bus i should take to go from 300 feet to Badda	Bus available on this route: BRTC
Which Bus i should take to go from 300 feet to Bakshi Bazar	Bus available on this route: BRTC Bus available on this route: BRTC
Which Bus i should take to go from 300 feet to Banani	Bus available on this route: BRTC
Which Bus i should take to go from 300 feet to Bangla College	Bus available on this route: BRTC Bus available on this route: BRTC
Which Bus i should take to go from 300 feet to Bangla Motor	Sorry, No bus on this route Sorry, No bus on this route
Which Bus i should take to go from 300 feet to Bashtola	Sorry, No bus on this route
Which Bus i should take to go from 300 feet to Bashundhara	Bus available on this route: BRTC Sorry, No bus on this route
Which Bus i should take to go from 300 feet to Bata Signal	Sorry, No bus on this route
Which Bus i should take to go from 300 feet to Bhulta	Bus available on this route: BRTC
Which Bus i should take to go from 300 feet to Bijoy Nagar	Sorry, No bus on this route
Which Bus i should take to go from 300 feet to Bijoy Sarani	Bus available on this route: BRTC Sorry, No bus on this route

Table 4.1.2.2 : Final conversational training data

DS1	FDS
categories: - route conversations: - - Which Bus i should take to go from 300 feet to Aarong - Bus available on this route BRTC - - Which Bus i should take to go from 300 feet to Abdullahpur - Bus available on this route BRTC - - Which Bus i should take to go from 300 feet to Adabor - Sorry, No bus on this route - - Which Bus i should take to go from 300 feet to Adamjee College - Sorry, No bus on this route - - Which Bus i should take to go from 300 feet to Agargaon - Sorry, No bus on this route - - Which Bus i should take to go from 300 feet to Airport - Bus available on this route BRTC - - Which Bus i should take to go from 300 feet to Amin Bazar - Bus available on this route BRTC - - Which Bus i should take to go from 300 feet to Ansar Camp - Bus available on this route BRTC	categories: - route conversations: - - From 300 feet to Aarong - Bus available BRTC - - From 300 feet to Abdullahpur - Bus available BRTC - - From 300 feet to Airport - Bus available BRTC - - From 300 feet to Amin Bazar - Bus available BRTC - - From 300 feet to Ansar Camp - Bus available BRTC - - From 300 feet to Arambagh - Bus available BRTC - - From 300 feet to Asad Gate - Bus available BRTC - - From 300 feet to Azompur - Bus available BRTC - - From 300 feet to Badda - Bus available BRTC - - From 300 feet to Banani - Bus available BRTC - - From 300 feet to Bangla College - Bus available BRTC

Table 4.1.2.3 : Dataset patterns to build different dataset

DS1 Pattern	DS2 Pattern	FDS pattern
1. Which bus i should take to go from A to B 2. I want to go from A to B 3. Show me the bus list from A to B 4. How I go from A to B 5. Tell me the route from A to B 6. Bus list from A to B 7. Available bus list from A to B 8. Available buses from A to B 9. A to B 10. From A to B 11. How can I go from A to B 12. Will you tell me the route from A to B 13. Will you tell me bus list from A to B 14. Please tell me bus list from A to B 15. I want to go B from A 16. How I go B from A	1. From A to B 2. From B to A	1. From A to B 2. From B to A

4.2 Data Training

4.2.1 SpaCy

SpaCy is a python library used for advanced natural language processing. In the market it is famous as “Industrial Natural Language Processing in Python”.

There is another library known as CyCon which was tremendously used in SpaCy. It was used to improve the related modules. It is mostly used at industrial level. It has some specialty building things in the industry.

4.2.2 Training

Qodra chatbot is a goal based closed domain chatbot. It will be able to answer about the bus names and some other basic conversation. Chatterbot library has a way to train the bot instance with the corpus data and other data formats. At first it's instance has no knowledge about the outside world. Then these data that was defined to train ,goes through a neural network and counts the frequency of these words, there are some related algorithms specified to tweak the value of training data. The search algorithm is known as **levenshtein distance** used to predict the confidence value of the users input.

levenshtein distance algorithm takes two strings as input and compares the distance between two strings. Finds out the best match value and returns as the confidence value.

Some python program was written to begin the training session. First corpus datas is stored in the project directory and in the chatterbot instance it was defined its location and it's name. Then just execute the python file and instance will make the knowledge database to store the training. It is not necessary to train every time when the program is running next time. What we can do , we can use Pickle to store the knowledge database. Basically it saves the model and for the next time it uses the previous knowledge database. Unless any changes made to the database file it won't retrain the database.

4.3 Training Storage Database

Since we thought to deploy the bot online, so decided to use a traditional database. We implemented the bot into a django web application .

There we used the mysql database. First we wrote a training program to train the database. These training results are stored on a single sqlite.db file. A screenshot is given below - how the internal structure is made inside of the database file.

Id	Text	Search_text	Conversation	Created_at	In_response_to	Search_in_respons...	Persona
1	From 300 feet to Aar...	IN:300 CD:vertebrate...	training	2020-02-19 10:48:15...	Null		
2	Bus available BRTC	NNP:available JJ:brtc	training	2020-02-19 10:48:15...	From 300 feet to Aar...	IN:300 CD:vertebrate...	
3	From 300 feet to Abd...	IN:300 CD:vertebrate...	training	2020-02-19 10:48:15...	Null		
4	Bus available BRTC	NNP:available JJ:brtc	training	2020-02-19 10:48:15...	From 300 feet to Abd...	IN:300 CD:vertebrate...	
5	From 300 feet to Airp...	IN:300 CD:vertebrate...	training	2020-02-19 10:48:15...	Null		
6	Bus available BRTC	NNP:available JJ:brtc	training	2020-02-19 10:48:15...	From 300 feet to Airp...	IN:300 CD:vertebrate...	
7	From 300 feet to Ami...	IN:300 CD:vertebrate...	training	2020-02-19 10:48:15...	Null		
8	Bus available BRTC	NNP:available JJ:brtc	training	2020-02-19 10:48:15...	From 300 feet to Ami...	IN:300 CD:vertebrate...	
9	From 300 feet to Ans...	IN:300 CD:vertebrate...	training	2020-02-19 10:48:15...	Null		
10	Bus available BRTC	NNP:available JJ:brtc	training	2020-02-19 10:48:15...	From 300 feet to Ans...	IN:300 CD:vertebrate...	
11	From 300 feet to Ara...	IN:300 CD:vertebrate...	training	2020-02-19 10:48:15...	Null		
12	Bus available BRTC	NNP:available JJ:brtc	training	2020-02-19 10:48:15...	From 300 feet to Ara...	IN:300 CD:vertebrate...	
13	From 300 feet to Asa...	IN:300 CD:vertebrate...	training	2020-02-19 10:48:15...	Null		
14	Bus available BRTC	NNP:available JJ:brtc	training	2020-02-19 10:48:15...	From 300 feet to Asa...	IN:300 CD:vertebrate...	
15	From 300 feet to Azo...	IN:300 CD:vertebrate...	training	2020-02-19 10:48:15...	Null		
16	Bus available BRTC	NNP:available JJ:brtc	training	2020-02-19 10:48:15...	From 300 feet to Azo...	IN:300 CD:vertebrate...	
17	From 300 feet to Bad...	IN:300 CD:vertebrate...	training	2020-02-19 10:48:15...	Null		
18	Bus available BRTC	NNP:available JJ:brtc	training	2020-02-19 10:48:15...	From 300 feet to Bad...	IN:300 CD:vertebrate...	

Figure 4.3.1 : Database table structure in SQL database

4.4 Response selection

Responses are selected on chatbot based on the training. If the dataset is good and efficient enough then the response will be efficient. Qodra chatbot is trained with about 57 thousand responses. A confidence value is generated with the output response. Some sample responses are shown below compared to the other chatbot.

Table 4.4.1 : Responses of Qodra chatbot of different questions

Question	Final Answer	Generated Answer	Confidence
What is two plus two?	I don't understand your question!!	From Mirpur 1 to Bangla College	0.00
What is the color of a yellow car?	I don't understand your question!!	Bus available Akash	0.00
What is the meaning of life?	I don't understand your question!!	There are other entities who are better versed on the topic.	0.24
What are your interests?	I am interested in all kinds of things. We can talk about anything!	I am interested in all kinds of things. We can talk about anything!	0.98
What is your number?	I don't have any number	I don't have any number	0.97
What is a computer?	So you think i am a machine. What characteristics made you think so?	So you think i am a machine. What characteristics made you think so?	1.00
from khilkhet to new market	Bus available VIP	Bus available VIP	1.00
from new market to housebuilding	Bus available Bkash	Bus available Bkash	1.00

Table 4.4.2 : Responses of NCM Chatbot of different questions [1]

User	NCM chatbot
what is two plus two?	Uh, well, we got four
what is the color of a yellow car?	yellow
what is the meaning of life?	i mean, what is life?
What are your interests?	I like reading and writing.
What is your number?	My name is AnneMeansGrace. What is your name?
What is a computer?	What do you mean?

NCM is an open domain neural network based chatbot where Qodra is a closed domain chatbot, who can make basic conversations too.

CHAPTER 5
TESTING AND RESULT

5.1 Testing

5.1.1 Manual Testing

After completing the random testing we manually tested the bot for the result. We tweaked some parameters of different response functions to get the best value. In most cases we achieved our target to obtain the result.

5.1.2 Random Testing

Random testing was the most beautiful part of this project. We took 100 random dataset of route data & stored in a text file. After completing the training , we automated that text file to generate responses. Then we matched the generated response with the actual answer. We obtained about 76% of accuracy of generating the correct response. Test dataset are given below.

Table 5.1.2.1 : Dataset Testing of Qodra.

Questions	Actual Answers	Qodra's Answers
From 300 feet to Palli Bidyut	Bus available BRTC	Bus available BRTC

From Aarong to Nilkhet	Bus available Ashirbad Paribahan	Bus available Ashirbad Paribahan
From Abdullahpur to New Market	Bus available VIP	Bus available VIP
From Agargaon to Narayanganj	Bus available Himachol	Bus available Himachol
From Airport to Nabisco	Bus available Balaka	Bus available Balaka
From Amin Bazar to Motsho Bhobon	Bus available Al Madina Plus One	Bus available Al Madina Plus One
From Ansar Camp to Mohammadpur	Bus available BRTC	Bus available BRTC
From Arambagh to Mohakhali	Bus available 6 No	Bus available 6 No
From Asad Gate to Moghbazar	Bus available Labbaik	Bus available Labbaik
From Ashulia Bazar to Mirpur 2	Bus available Alif	Bus available Alif
From Azimpur to Mirpur 14	Bus available Grameen	Bus available Bihanga
From Badda to Mirpur 12	Bus available Achim Paribahan	Bus available Achim Paribahan
From Banani to Mirpur 10	Bus available Al Makka Transport	Bus available Al Makka Transport
From Bangla College to Mirpur 1	Bus available Achim Paribahan	Bus available Achim Paribahan
From Bangla Motor to Manik Mian Avenue	Bus available 8 No	Bus available 8 No
From Bashtola to Madhya Badda	Bus available Achim Paribahan	Bus available Achim Paribahan
From Bashundhara to MES	Bus available Achim Paribahan	Bus available Akik Paribahan

From Bata Signal to Link Road	Bus available BRTC	Bus available BRTC
From Bhulta to Labaid	Bus available Meghla Transport	Bus available Meghla Transport
From Bijoy Nagar to Kuril Bishwa Road	Bus available Akash	Bus available Akash
From Bijoy Sarani to Khilkhet	Bus available Alif	Bus available Alif
From Chairman Bari to Khamarbari	Bus available BRTC	Bus available BRTC
From City College to Katabon	Bus available 7 No	Bus available 7 No
From Dhakeshwari to Kalabagan	Bus available Bikalpa City Service	Bus available Bikalpa City Service
From Dhanmondi 15 to Kakrail	Bus available BRTC	Bus available BRTC
From Dhanmondi 27 to Kakoli	Bus available Bkash	Bus available BRTC
From Dhanmondi 6 to Jatrabari	Bus available Moumita Transport	Bus available City Link
From Dhanmondi 8 to Jashimuddin	Bus available Bkash	Bus available BRTC
From Dhour Chourasta to Japan Garden City	Bus available Alif	Bus available Alif
From Diabari to Janapath	Bus available Raida	Bus available Raida
From ECB Chottor to Jahangir Gate	Bus available Al Makka Transport	Bus available Al Makka Transport
From Elephant Road to Ittefaq	Bus available Shahria Enterprise	Bus available Shahria Enterprise
From Farmgate to Housebuilding	Bus available Bangabandhu Avi New	Bus available Bangabandhu Avi New

From Fulbaria to High Court	Bus available Bangabandhu Avi New	Bus available Bangabandhu Avi New
From Gabtoli to Gulshan Bridge	Bus available Agradut-Boishakhi	Bus available Agradut-Boishakhi
From Golap Sha Mazar to Gulshan	Bus available Bihanga	Bus available 7 No
From Gulshan to Golap Sha Mazar	Bus available Bihanga	Bus available 7 No
From Gulshan Bridge to Gabtoli	Bus available Agradut-Boishakhi	Bus available Agradut-Boishakhi
From High Court to Fulbaria	Bus available Bangabandhu Avi New	Bus available Bangabandhu Avi New
From Housebuilding to Farmgate	Bus available VIP	Bus available VIP
From Ittefaq to Elephant Road	Bus available Shahria Enterprise	Bus available Shahria Enterprise
From Jahangir Gate to ECB Chottor	Bus available Al Makka Transport	Bus available Al Makka Transport
From Janapath to Diabari	Bus available Raida	Bus available Raida
From Japan Garden City to Dhour Chourasta	Bus available Alif	Bus available Alif
From Jashimuddin to Dhanmondi 8	Bus available VIP	Bus available BRTC
From Jatrabari to Dhanmondi 6	Bus available Moumita Transport	Bus available BRTC
From Kakoli to Dhanmondi 27	Bus available Bkash	Bus available BRTC
From Kakrail to Dhanmondi 15	Bus available BRTC	Bus available Taranga Plus
From Kalabagan to Dhakeshwari	Bus available Bikalpa City Service	Bus available Bikalpa City Service

From Katabon to City College	Bus available 7 No	Bus available 7 No
From Khamarbari to Chairman Bari	Bus available BRTC	Bus available BRTC
From Khilkhet to Bijoy Sarani	Bus available Alif	Bus available Alif
From Kuril Bishwa Road to Bijoy Nagar	Bus available Akash	Bus available Akash
From Labaid to Bhulta	Bus available Meghla Transport	Bus available Meghla Transport
From Link Road to Bata Signal	Bus available BRTC	Bus available BRTC
From MES to Bashundhara	Bus available Achim Paribahan	Bus available Achim Paribahan
From Madhya Badda to Bashtola	Bus available Achim Paribahan	Bus available Achim Paribahan
From Manik Mian Avenue to Bangla Motor	Bus available 8 No	Bus available 8 No
From Mirpur 1 to Bangla College	Bus available Achim Paribahan	Bus available Achim Paribahan
From Mirpur 10 to Banani	Bus available Al Makka Transport	Bus available Al Makka Transport
From Mirpur 12 to Badda	Bus available Achim Paribahan	Bus available Achim Paribahan
From Mirpur 14 to Azimpur	Bus available Grameen	Bus available Bihanga
From Mirpur 2 to Ashulia Bazar	Bus available Alif	Bus available Alif
From Moghbazar to Asad Gate	Bus available Labbaik	Bus available Labbaik
From Mohakhali to Arambagh	Bus available 6 No	Bus available 6 No

From Mohammadpur to Ansar Camp	Bus available BRTC	Bus available BRTC
From Motsho Bhobon to Amin Bazar	Bus available Al Madina Plus One	Bus available Al Madina Plus One
From Nabisco to Airport	Bus available Ashulia Classic	Bus available Ashulia Classic
From Narayanganj to Agargaon	Bus available Himachol	Bus available Himachol
From New Market to Abdullahpur	Bus available Bkash	Bus available Bkash
From Nilkhet to Aarong	Bus available Ashirbad Paribahan	Bus available Ashirbad Paribahan
From Sign Board to 300 feet	Bus available BRTC	Bus available BRTC
From Shyamoli to Aarong	Bus available 7 No	Bus available 7 No
From Shukrabad to Abdullahpur	Bus available Bkash	Bus available Bkash
From Shewra to Agargaon	Bus available Alif	Bus available Alif
From Shainik Club to Airport	Bus available Bangabandhu Avi New	Bus available Bangabandhu Avi New
From Shaheen College to Amin Bazar	Bus available Agradut-Boishakhi	Bus available Agradut-Boishakhi
From Shahbag to Ansar Camp	Bus available BRTC	Bus available BRTC
From Science Lab to Arambagh	Bus available BRTC	Bus available BRTC
From Ray Shahed Bazar to Azimpur	Bus available Bihanga	Bus available Bihanga
From Rampura Bridge to Azompur	Bus available Akash	Bus available Dewan

From Rampura to Badda	Bus available Achim Paribahan	Bus available Achim Paribahan
From Prottasha to Banani	Bus available Bkash	Bus available Bkash
From Press Club to Bangla College	Bus available BRTC	Bus available BRTC
From Paltan to Bashtola	Bus available Akash	Bus available Akash
From Palli Bidyut to Bashundhara	Bus available BRTC	Bus available BRTC
From Nilkhet to Bata Signal	Bus available Dewan	Bus available BRTC
From Naya Bazar to Bijoy Nagar	Bus available Akash	Bus available Akash
From Natun Bazar to Bijoy Sarani	Bus available Agradut-Boishakhi	Bus available Agradut-Boishakhi
From Motsho Bhobon to Chairman Bari	Bus available Bangabandhu Avi New	Bus available Bangabandhu Avi New
From Mohakhali to Chiriakhana	Bus available Modhumoti Paribahan	Bus available Modhumoti Paribahan
From Mirpur 2 to Dhaka EPZ	Bus available BRTC	Bus available BRTC
From Mirpur 12 to Dhanmondi 27	Bus available Bihanga	Bus available BRTC
From Mirpur 11 to Dhanmondi 32	Bus available Bihanga	Bus available BRTC
From Mirpur 10 to Dhanmondi 6	Bus available Bahon	Bus available Bahon
From Mirpur 1 to Dhanmondi 8	Bus available Bahon	Bus available BRTC
From Manik Mian Avenue to Dhour Chourasta	Bus available Bkash	Bus available Bkash

From Madhya Badda to Diabari	Bus available Akash	Bus available Akash
From MES to ECB Chottor	Bus available Achim Paribahan	Bus available Achim Paribahan
From Link Road to Elephant Road	Bus available BRTC	Bus available BRTC

Explain How you use the above test data set. Which code you have used and how you measured the accuracy is 76%. Provide details.

CHAPTER 6
DEPLOYMENT TO PRODUCTION

6.1 Django Integration

Django is a python framework for developing web technologies. Since our project is solely based on the python language, we wanted to deploy the inside of a web technology. There is also a similar framework available known as Flask. But for PIP support we used the django framework. The python 3.7 version is used for developing the project & the Django version is 3.2. Since servers cost a lot of money, we decided to deploy it in our local personal computer and be able to make the IP universal by route forwarding technique. Then making the local machine the server machine to act as an API. That will also be used to serve the static files.

Our prototype is nice for console integration. But we took the application to the next level. We build an API to communicate with the user's request. Then we build a static web page with a nice interface which will interact with users. We also build the mobile application which is also able to communicate with the API prototype.

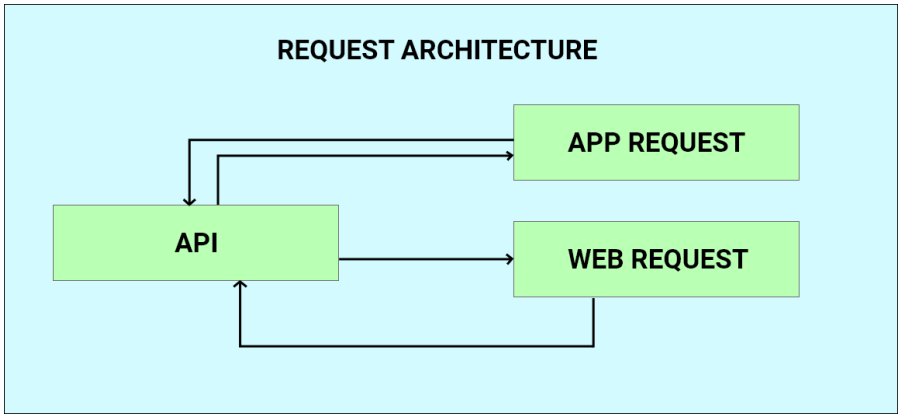


Figure 6.1.1 : Installation of Django

First we have to install python 3.7 on the target machine. Then we have to create a virtual environment. The main facilities give us about venv is it isolates the projects environment from the system environment. To create the virtual environment we have to use the following command.

```
python3 -m venv /path/to/new/virtual/environment
```

To activate the virtual environment we have to use the following command.

Platform	Shell	Command to activate virtual environment
POSIX	bash/zsh	$ source <venv>/bin/activate
	fish	$. <venv>/bin/activate.fish
	csh/tcsh	$ source <venv>/bin/activate.csh
	PowerShell Core	$ <venv>/bin/Activate.ps1
Windows	cmd.exe	C:\> <venv>\Scripts\activate.bat
	PowerShell	PS C:\> <venv>\Scripts\Activate.ps1

Figure 6.1.2 : Command to activate virtual environment (venv) [5]

Since our target machine is windows based, we used the activate.bat to activate the venv. After activating the virtual environment we have to install Django. To install django and other related libraries we used the following command.

```
pip install django
pip install Pillow==2.2.2
pip install chatterbot
pip install django chatterbot
pip install chatterbot-corpus
```

By using these commands a lot of dependency libraries will be installed. Then we build django apps to integrate our prototype into web applications. Before that the web interface was converted into HTML/CSS, JAVASCRIPT.

Here javascript helped us to move the user request form frontend to backend and brings the response. Static files must be loaded into the django application. Following commands are used to complete the setup.

```
django-admin startproject mysite # To create a new project
py manage.py runserver # To activate the server
py manage.py collectstatic # To collect static file like html files,
py manage.py createsuperuser #To access the admin panel
```

This is how the django installation procedure is completed.

6.1.2 Setting up the Qodra configuration

To set up the chatbot into a django app needs to tweak some parameters. We need to specify the database path and configure it with the train.py file. train. py file is the core file which takes input from the predefined specified location and saves to the shown and selected database. We need to add some additional code to the core settings.py file of the project app we are using. Additional codes are given below.

settings.py

```
import json
from django.views.generic.base import TemplateView
from django.views.generic import View
from django.http import JsonResponse
from chatterbot import ChatBot
from chatterbot.ext.django_chatterbot import settings

chatterbot = ChatBot(**settings.CHATTERBOT)
```

train.py

```
chatbot = ChatBot('Chatterbot',
trainer='chatterbot.trainers.CorpusTrainer',
storage_adapter='chatterbot.storage.SQLStorageAdapter',
database_uri='sqlite:///database.db',
response_selection_method=get_first_response,
```

```
logic_adapters=[
        {

             "import_path": "chatterbot.logic.BestMatch",
            "import_path": "chatterbot.logic.MathematicalEvaluation",
            "statement_comparison_function":
"chatterbot.comparisons.levenshtein_distance",
            "response_selection_method":
"chatterbot.response_selection.get_first_response",

                    }
    ]
)

trainer = ChatterBotCorpusTrainer(chatbot)
```

This additional code should be included to tweak the Qodra chatbot to fetch out the best result. We found these parameters perfect by testing multiple tests based on different parameters of different results.

After configuring these training and settings files some modules need to be written to communicate or handle requests from the API. Those modules are written inside views.py file. After that URL routing configuration was set up with these modules. Now the whole system is ready. First we have to run the train.py file and later run the server into 0.0.0.0 in 8000 port. The API should be live to and running.

6.2 Deploy to a remote location

Deploying a project in python is an easy task but it isn't free to deploy the project of a machine learning project with a large database. Since we couldn't find a suitable server to deploy our project , we made our personal machine to act as a server.

Now the interesting tricks we used to act our personal machine as an API we configure the router to control the hits from the user. More easily we can say that we connected our personal machine to the internet, now this machine is a part of the web directory. Anyone can access from the internet to my local machine. That machine will serve static files like

HTML/CSS templates, images and other media data. These files will be served by the django application which was already configured and ready for generating responses.

Router configuration was an important part to make our local machine to act as an API and server machine. The way we were able to do this is known as port forwarding. Every router should have that option. We used a TP-LINK router. From that access point we told the router that if any request comes to that ip address and if the port number is 8000 then forward the request to the computer IP address. One more important fact is that a router must have the real IP. That's how we were able to configure the route and deploy the project into a remote location.

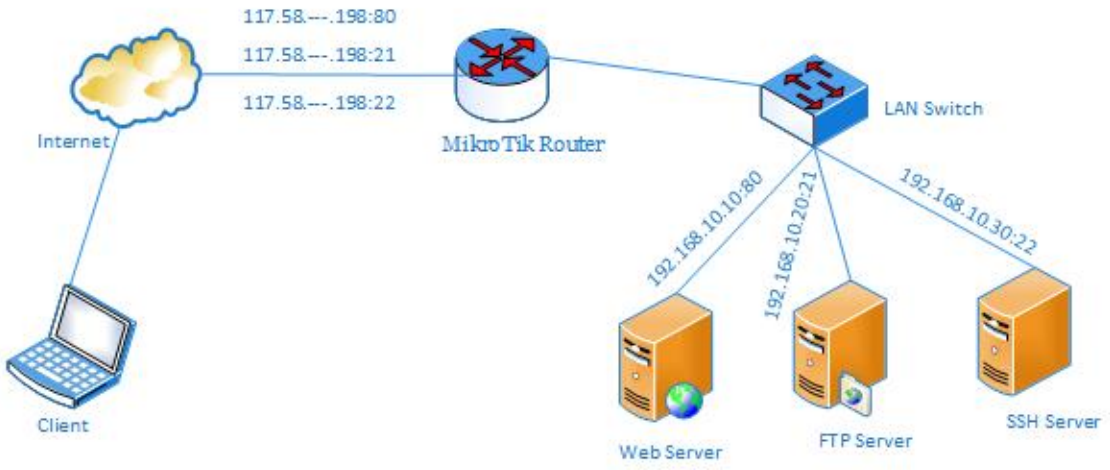


Figure 6.2.1: The concept of router port forwarding [5]

If we wanted to deploy that django application into a highly configured remote server we would have to maintain some steps to deploy the software. At first we would need the ubuntu server or window server operating system. Based on the operating system, we would have to install our environment packages, then we would have to install **nginx** to

keep the server live. Next we would need **gunicorn** to maintain some protocols, then upload the project files and configure the ip address and routes.

CHAPTER 7
ANDROID APPLICATION

7.1 Introduction

Android is a Linux based operating system. It is designed for touch screen mobile devices such as smartphones and tablet computers. Android is one of the most widely used mobile operating systems in the 20th century. The android software that was founded in Palo Alto of California in 2003.

Android applications are more comfortable and advanced for the users for the portability. The hardware that holds on android software is based on the ARM architecture platform. The android is an open source OS which means that it's free and anyone can use it.

7.2 Purpose

Our purpose behind developing the android application is to reach out users. Today's generation is fond of smartphones. They spend most of the time using smartphones. Often they are unable to choose what bus they should take to reach their destination. So we are trying to give them something which will help them to solve this problem. They can find the solution in their smartphones which they carry all the time. They can talk with something to solve their problem.

7.3 Requirements

Before getting hands on the development of the mobile application we must have to fulfill the requirements. Here is the list of the software requirements dependency.

- Windows / Linux / MAC
- JAVA
- Android Studio
- Android 4.0.3 (API level 15) (minimum)

7.4 Dependency Requirements

- AppCompat
- ConstraintLayout
- Junit

- RecyclerView
- Volley

7.5 User Permission Requirements

- INTERNET
- RECORD_AUDIO

7.6 Architectural Design

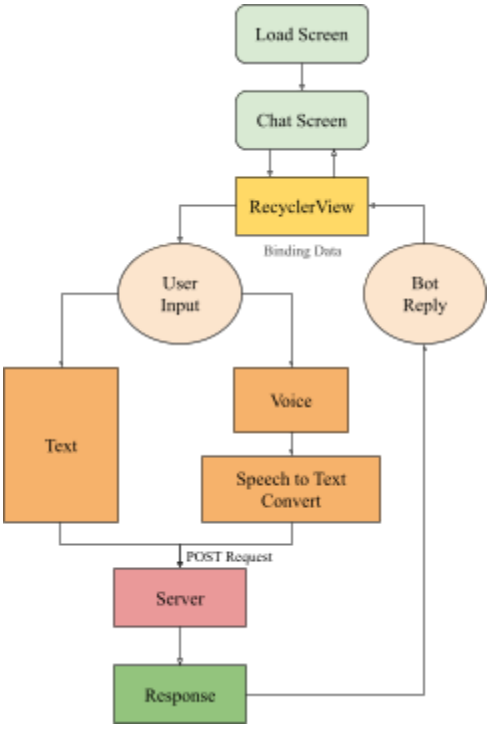


Figure 7.6.1: Android Application Process

7.7 Implementation

In order to implement the application we need to create some pages. In android it is known as activity. Behind those activity XML files we create some JAVA classes in order to maintain those XML layouts.

While building the application we had to go through some dependency softwares and platforms. Like Server Communication, Static Pages, Dynamic View, Response time, Speech Recognition.

7.7.1 Static Pages

In the application there were some static pages like our FAQ and Home Screen pages. Here, we statically placed our required data like some manual documentation, how to use the application, how this application works, our information etc. We pass this information via the Intent class of JAVA. Also we use Timer class to jump into other pages after a certain amount of time.

Intent

An Intent may be a simple message object that's accustomed to communicate between android components like activities, content providers, broadcast receivers and services. Intents also are accustomed to transfer data between activities. Intents are used generally for starting a brand new activity using `startActivity()`.

Timer

A facility for threads to schedule tasks for future execution in an exceedingly background thread. Tasks are additionally scheduled for one-time execution, or for repeated execution at regular intervals.

This class is thread-safe. You can handle multiple threads with one single Timer object without any kind of external synchronization.

7.7.2 Dynamic View

We had to do some dynamic operations in order to view our chat data into application. Here we use RecyclerView to dynamically arrange our data in Bottom to Top view. We build a custom logic adapter to bind our data into LayoutInflater.

Our logic adapter grabs data from the user and passes it through the web server via POST method. We set some conditions so that we can define our data where it comes from.

RecyclerView

RecyclerView is a sophisticated version of ListView and GridView. It has some improved performance and other benefits. It's more flexible and efficient than ListView and GridView.

RecyclerView is a ViewGroup. It helps to display both vertical and horizontal scrolling lists of elements. It supports an outsized set of information items.

It's a widget of the `android support-v7` version.

LayoutInflater

This class is used to represent layout XML files into its corresponding View objects. It takes an XML file as input. It builds the View objects from it.

We use the Inflater keyword in the corresponding layout(xml) file. This keyword is used in the `onCreate()` function of the class.

There are two ways to create a UI in android. One is static and another is dynamic or programmatically. LayoutInflater is mainly used for dynamic layout.

7.7.3 Server Communication

As you know, Our application directly communicates with our web server which is built on Python 3.7. How do we do this? How does this work?

In order to communicate with our web server, we have gone through some packages or classes of Java. Like Volley, Uri, Request, RequestQueue, Response, StringRequest, JSONObject

We grab data from the application and wrap it through the Volley StringRequest method. Volley returns a response in JSONObject. We parse the JSONObject into our application activity.

Volley

Volley is an HTTP library that creates networking very easy and fast, for Android apps. It was developed by Google and introduced during Google I/O 2013. It manages the processing time and caching of network requests. It helps us to write the same network code again and again. It's not suitable for streaming operations or giant download because it holds all responses in memory during parsing.

Volley populates a UI using RPC-type operations, such as fetching a page of search results as structured data. It easily integrates with any protocol and comes out of the box with support for raw strings, images, and JSON. It provides built-in support for the features you need, Volley frees us from writing boilerplate code and allows us to concentrate on the logic that is specific to our app.

URI - Uniform Resource Identifier

Uniform Resource Identifier (URI) is a string of characters which is used to identify a resource. A URI identifies a resource either by a name, or a location, or both. Such identification helps to enable interaction with representations of the resource over a stable network, like the World Wide Web, with help of a specific protocol.

Beside some minor changes, an instance of this class represents a URI defined by RFC 2396: Generic Syntax, Uniform Resource Identifiers (URI): amended by RFC 2732: Format for Literal IPv6 Addresses in URLs. The Literal IPv6 address format also supports scope_ids. The usage and syntax of scope_ids is described here. This class

provides constructors for creating URI instances from their components or by parsing their string forms, methods for normalizing, resolving, and relativizing URI instances and methods for accessing the various components of an instance. Instances of this class are immutable.

The main and key operations of this class are those of normalization, resolution, and relativization.

Request

Requests do the parsing of raw responses and Volley takes care of dispatching the parsed response back to the most thread for delivery.

At a high level, we use Volley to create a RequestQueue and pass it through Request objects. The RequestQueue handles the worker threads for running the network operations, parsing responses, and reading from and writing to the cache. Requests does the parsing of raw responses and Volley takes care of dispatching the parsed response back to the main thread for delivery.

That's how we send a request using the `Volley.newRequestQueue` convenience method, which sets up a RequestQueue.

RequestQueue

RequestQueue is used to handle your cache and stack your request. We need to create RequestQueue in our application class. So that only we can use the same requestQueue from multiple activities.

A RequestQueue needs multiple things to do its job. A cache to handle caching and a network to perform transport of the requests.

There are some standard implementations of these available in the Volley. DiskBasedCache supplies a one-file-per-response cache with an in-memory index, and BasicNetwork supplies a network transport based on our preferred HTTP client.

Volley's default network implementation is BasicNetwork. It must be initialized with the HTTP client of our app which is used to connect to the network. Typically this is an HttpURLConnection.

Response

Represents the successful results of invoking an API method in Google Play services employing a subclass of GoogleAPI. Wraps an instance of a Result.

StringRequest

HTTP Request where the response is parsed as a String. It specifies a URL and receives a raw string in response.

In order to send a request we have predefined what type of request we are sending. Like POST, GET, PUT, DELETE. We need to specify our Response Listener method in order to receive our response.

After that an Error Listener method in case we get some error while sending a request. With those we have to send a parameter which will obtain our string as input.

JSON

JSON stands for JavaScript Object Notation. It is lightweight, structured, human readable and easy to parse. It's a best alternative to XML when our android app needs to interchange data from the server. XML parsing is very complex as compared to JSON parsing.

JSONObject

A JSONObject represents the data in the key and value pair. Names are non-null unique strings.

7.7.4 Response time

HTTP library makes networking very easy and fast. If our network connection is suitable we got our response in less than 1 second.

If we face a bad network or any kind of issue in sending a POST request then we have to go through a RetryPolicy. Which helps our application to wait until 50000ms. Then it shows an error message and goes for the next conversation.

7.7.5 Speech Recognition

Our application can recognize our speech through our smartphones default language model. In order to do this we go through the SpeechRecognizer class of JAVA.

SpeechRecognizer

This class supplies access to the speech recognition service. This service grants access to the speech recognizer. The implementation of this API is likely to stream an audio to remote servers to carry out speech recognition. This API is not intended to be used for continuous recognition, which would consume a significant amount of bandwidth and battery.

The application must have `android.Manifest.permission#RECORD_AUDIO` permission.

7.8 Importance of Android Application

The main principle of developing an android application is it is portable and user friendly. As you know our main goal is helping people to find the right bus inside Dhaka city. If

you are on the road you will not carry your desktop. It's quite impossible. That's why we need an operating system which we can use anywhere we want. Thanks to our technology now this is possible. We have smartphones, tablet pc which we can carry with us.

As you know, around 70 percent of people use Android as their smartphones operating system. We can easily share our idea or application through people via Android. So, you don't have to be confused about which bus you will take in order to travel around Dhaka city.

7.9 Improved version of android application

This application fetches answers from a web server. When we hit too many requests on server, server gets stuck and our application is unable to fetch response from server. We need to improve our server response in order to fetch our answers.

Each refresh we lost our previous conversations. Which is quite a problem we are facing. We are trying to fix that problem.

CHAPTER 8
IMPROVEMENT & CONCLUSION

8.1 Response Improvement

In Qodra chatbot we tried to generate responses on route conversation. It can also produce some basic human conversation at the introductory level. Qodra chatbot acts in a closed domain manner. Right now we would only be able to build a route dataset on Dhaka city which is in English language. Our Dataset is limited to Dhaka city. If we are able to manage or build a dataset of other cities we would be able to produce much better results. To work with a large amount of dataset we would need a highly configured computer or server computer which can deal with large amounts of dataset.

Our next intention would be to build a more reliable dataset. Current version of the dataset may have some old unreliable data. Most of the dataset is collected from different types of web applications, mobile applications and some other sources. Then we take all these datas and clean the dataset as much as possible. After that corpus data is produced by a C++ program.

The next step would be to store the user data. User data is the most important feedback to understand and improve the product at the best level. By storing specific user data we would be able to understand the needs of the users.

Our next improvement would be to add an optional feature. Images will be fed to the bot to understand the object. Then after showing related images to bot may help to predict the object. The best use would be to build the bot for disease prediction. Specific applications are being developed to predict different types of disease. Such bot can help in these cases.

Our ultimate goal is to work with the Bangla language. First we would need to build the

dataset in bangla then we have to convert it to corpus dataset. The difficult part is to train the bot with a corpus dataset in bangla language and getting the expected response. This is our ultimate goal to achieve the milestone with bangla language and also improving the response.

We thought about a simple feature known as weather forecasting. It is an interesting fact that when people will search for how to go from that place to that place it means they are going outside. It would be nice to give some extra information such as take an umbrella with you, it's cloudy today.

We are thinking of making these improvements.These are really interesting and important improvements to make.

8.2 Limitations

The first thing to build a machine learning project is data. The data wasn't enough. Since we are developing a prototyping project we had some problems to solve. We had some limitations that we couldn't cross. We couldn't make our bot communicate in bangla because of the lackings in bangla dataset. Our chatbot is able to communicate only in english. Since our chatbot is goal based so it cannot talk randomly in any topic such as google assistant, siri or alexa.

However, for many complex sentences it may produce wrong answers or default response. It is slower in generating response. Sometimes it can be time consuming depending on the server, new response or old response, Internet connections etc. If the server computer is slower than it'll take time to generate a response.

Our future plan is to eliminate those obstacles to achieve a greater result.

8.3 Conclusion

Our resources are limited , by adding a simple feature from time to time it can turn into a tremendous project with a large database. We took a problem and tried to solve the first

milestone. More features can be added to make the bot more powerful and more useful. People go from one place to another. They face a situation when they go to a new place and don't know how to get from one place to another. They don't know the running vehicles in those areas. In that case people ask other people sometime they get misleading information from people. This is a very tedious and time killing process. So we tried to solve that problem in a very interesting way.

Qodra project is introduced with the chatbot development project in a specific manner. Based on the Qodra bot platform , algorithms it can respond to the user with the routing answer.

References

Conference/Journal Papers:

[1] http://dspace.bracu.ac.bd/xmlui/handle/10361/8122

Image Contents:

[1] https://www.kdnuggets.com/2018/05/pytorch-tensor-basics.html
[2] https://www.nltk.org/
[3] https://www.researchgate.net/figure/A-simplified-WordNet-hierarchy-of-synsets_fig2_228707772
[4] https://docs.python.org/3/library/venv.html
[5] https://systemzone.net/mikrotik-port-forwarding-using-winbox/

Plagiarism Result

162-15-741

ORIGINALITY REPORT

13%	12%	3%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	4%
2	forum.androindian.com Internet Source	2%
3	cr.openjdk.java.net Internet Source	1%
4	www.geeksforgeeks.org Internet Source	<1%
5	Submitted to Nottingham Trent University Student Paper	<1%
6	abhiandroid.com Internet Source	<1%
7	developer.android.com Internet Source	<1%
8	Kumar Saurav, Seul Jung. "Robust Detection of the Coordinate of a Solar Panel Using Deep Learning Algorithm", 2019 19th International Conference on Control, Automation and	<1%

	Systems (ICCAS), 2019 Publication	
9	stackoverflow.com Internet Source	<1%
10	Krzysztof Stępień. "MANAGEMENT AND CONTROL OF SMART CAR WITH THE USE OF MOBILE APPLICATIONS", Information System in Management, 2017 Publication	<1%
11	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%
12	Submitted to The Scientific & Technological Research Council of Turkey (TUBITAK) Student Paper	<1%
13	Submitted to City University Student Paper	<1%
14	examples.javacodegeeks.com Internet Source	<1%
15	media.readthedocs.org Internet Source	<1%
16	en.wikipedia.org Internet Source	<1%
17	www.coursehero.com Internet Source	<1%

18	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1%
19	Submitted to University of Hertfordshire Student Paper	<1%
20	Submitted to Laureate Education Inc. Student Paper	<1%
21	Submitted to Pearson College Student Paper	<1%
22	Pitrat. "Around the Conscience", Artificial Beings, 01/01/2009 Publication	<1%
23	www.elprocus.com Internet Source	<1%
24	Submitted to University of California, Los Angeles Student Paper	<1%
25	Adrian Holovaty, Jacob Kaplan-Moss. "The Definitive Guide to Django", Springer Science and Business Media LLC, 2008 Publication	<1%
26	Eric Hsiao-Kuang Wu, Chun-Han Lin, Yu-Yen Ou, Chen-Zhong Liu, Wei-Kai Wang, Chi-Yun Chao. "Advantages and Constraints of a Hybrid Model K-12 E-Learning Assistant Chatbot", IEEE Access, 2020	<1%

	Publication	
27	Submitted to The University of Manchester Student Paper	<1%
28	Md. Kowsher, Farhana Sharmin Tithi, M Ashraful Alam, Mohammad Nurul Huda, Mir Md Moheuddin, Md. Golam Rosul. "Doly: Bengali Chatbot for Bengali Education", 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), 2019 Publication	<1%
29	Submitted to Innopolis University Student Paper	<1%
30	basketball.dailyherald.com Internet Source	<1%

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		