IMPLEMENTATION OF A BANGLA CHATBOT

A thesis submitted in partial fulfillment for the
Bachelor in Computer Science and Engineering

Tasnim Dewan Orin
ID-13101224

Supervised By:
Dr. Mahbub Alam Majumdar
Professor

Department of Computer Science and Engineering
BRAC University

April 2017

DECLARATION

We, hereby declare that this thesis is based on implementation we have done ourselves. Materials of work found by other researcher or open source library used are mentioned by reference. This thesis, neither in whole or in part, has been previously submitted for any degree.

Signature of Supervisor **Signature of Author**

_____ _____

Dr. Mahbub Alam Majumdar
Professor
Department of Computer Science and Engineering
BRAC University

Tasnim Dewan Orin
ID- 13101224

ABSTRACT

We propose a fully data driven retrieval based closed domain chatbot which can converse in Bengali with the user in a chat interface based on its knowledge base and through learning from interactions with the user. Bengali is the state language of Bangladesh and fourth most popular language in the world. Although the first chatbot Eliza was invented in 1964, the author is not aware of any other work where Bengali chatbot has been implemented. So it is the demand of time to build a chatbot in Bengali. This paper proposes the first Bengali chatbot named Golpo based on a language-independent natural language processing library with a learning mechanism. The experiment shows that our chatbot is able to give responses to the user in real time. At first, it matches the input with the existing queries in the database, then it calculates a confidence score for each matching sentences with the input, and finally, it selects the one with highest confidence score as the response to the input. We hypothesize that this implementation will help to build different goal oriented chatbot in Bengali, for example, customer care representatives, FAQ (Frequently Asked Questions) chatbot, online sales agent etc. Besides, one of the main contributions of this work is that Golpo will be able to provide a Bengali Corpus for research purpose in future. Experimental results show that Golpo outperforms the state-of-the-art model named Cleverbot (Saenz, 2010), and achieves reasonably good results when compared to Neural Conversational Model (NCM) (Vinyals, 2015), a generative based model. Based on the evaluation of users, we can say Golpo can produce syntactically correct and natural responses in Bengali.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to The Most Merciful Allah who blessed me with patience, determination, ability and opportunity to complete the thesis within time.

I would love to thank from the bottom of my heart to my supervisor Dr. Mahbub Majumdar who has accepted me to do thesis under his supervision and has helped me to complete it within time with his valuable guidance and advises. I am fortunate and grateful to be able to work under his supervision.

Finally, I am grateful to my family for believing in my ability and for supporting me through thick and thin. Also, I want to thank faculty members and staffs of the Department of Computer Science and Engineering, BRAC University for providing me with all kinds of help.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

1.1 Introduction

The absolute goal of the Artificial Intelligence research is to build a machine which can converse with a human such that no one can differentiate it from a real human being. After Alan Turing proposed his Turing Test in 1950 in his famous work "Computing Machinery and Intelligence" (Turing, 1950), it has been almost 60 years that researchers are trying to pass the test. As a part of the research to develop an intelligent conversational agent, in 1964, Eliza (Weizenbaum, 1966), a simulation of a Rogerian psychotherapist, was developed in Massachusetts Institute of Technology (MIT). It was capable of replying the sentences used by the users back to them. It was the beginning of the research on the conversational agent.

A conversational agent is a program which can converse in a natural language with the user either based on the knowledge base (retrieval based closed domain model) or by generating new sentences (generative based open domain model) in a chat interface and sometimes, it is able to perform actions based on the conversations (goal-oriented chatbots, for example, pizza ordering chatbot). It is popularly known as chatbot or chatterbot or bot.

A chatbot is one of the main concerns of the study of Human-Computer Interaction (HCI).We can cite the examples of Cleverbot[1] or Simsimi, automated tutorials and online assistants as chatbots in use. The use cases of chatbot include customer care representative, sales agent, FAQ answerers etc. With the rise of smartphones and other high computational devices, chatbot becomes one of the hot topics of Natural Language Processing (NLP) research. This is to be noted that all research were based on only one language which is English. Although some works have been done on Chinese and Spanish languages but there are very few in number. Other languages were left out because of the lack of quality data corpus and natural language

[1] http://www.cleverbot.com/

processing tools.

Bengali is the fourth most spoken language in the world. It is the mother tongue of the people of the Bangladesh and also the language of some part of India such as Calcutta. But is a matter of great regret that it is considered as low-density language. There are many languages considered to be low-density languages because insufficient digitized text material is available in the language even though millions of people speak the language (Islam, 2012). As a low-density language, there is no chatbot in Bengali. Therefore, this very notion motivates us to create a chatbot in Bengali.

Bengali Natural Language Processing (BNLP) is a recent concern of academic field. There is a lack of appropriate resources and a lot of work has to be done. Writing a perfect Chatbot is very difficult because it needs a very large database and must give reasonable answers to all interactions (Abdul-Kader, 2015). Due to unavailability of large Bengali conversation corpus, for the prototype, the system is provided with a sample hand-crafted dataset for general interactions. So the main objective of our work is to build a novel chatbot in Bengali which will able to interact in Bengali as well as building a corpus in Bengali. We name it “Golpo” which means “Story” in Bengali.

As chatbot will be as good as its knowledge base which matches the user’s input with the best-matched response in its database, due to lack of quality corpus in Bengali, the job becomes even more complicated. So for any researchers who wants to build a chatbot in Bengali, his/her first job will be to build an appropriate Bengali corpus. The main challenges of this research are finding an appropriate dataset to use as a knowledge base, dealing with misspelled inputs or grammatically incorrect sentences and being able to have an engaging human-like conversation. Since we do not find a suitable database for this purpose we manually prepared and annotated conversation corpus. In this work, for the implementation, we create an appropriate Bengali corpus from the scratch and design it in such a way that it is able to provide us with a generated Bengali corpus.

Since it is the first attempt so we are using the retrieval based system based on a pattern matching mechanism between the inputs and the hand crafted rules predetermined in the

knowledge base with a learning feature. Learning here means saving new phrases and then using them later to give appropriate answers for similar phrases (M. J. Pereira, 2013). The contribution of this paper is two folds. (1) We start the research of Bengali chatbot or conversational agent (2) Our system is able to provide a Bengali corpus which can be used for other Bengali Language Processing research in future.

We evaluate the chatbot based on user's satisfaction. Shawar, B. A., & Atwell, E. (2007, April) advised that "Evaluation should be adapted to the application and to user needs. If the chatbot is meant to be adapted to provide a specific service for users, then the best evaluation is based on whether it achieves that service or task". The main task of our research is to interact with the user in fluent and syntactically correct Bengali.

We compare Golpo with two popular English chatbots which are Neural Conversational Model (NCM) and the Cleverbot. In the experiments, our chatbot performs as well as the NCM. Cleverbot is quite similar to our work. Experimental evaluation shows our system outperforms Cleverbot in many cases. It is to be mentioned that our chatbot replies in syntactically correct Bengali.So we can state our chatbot is able to achieve its main goal. We believe our work will inspire a lot of researchers to come forward for the development of resources in Bengali Language Processing.

1.2 Objective

1. To create a Chatbot in Bengali which can take input from users in Bengali and also replies in Bengali.
2. To develop a Bengali corpus for training the chatbot and with the learning feature making the system being able to generate a corpus for future use.
3. The responses should be free from spelling and grammatical mistakes.
4. The responses should be consistent, coherent and in order.

1.3 Motivation

Last decade was the decade where a technological revolution took place. With the increase in usage of smartphones, the number of the social network users and mobile app users has increased manifold. Now the question is "What's next?". From recent research in the pattern of consumer behavior related to smartphones, it is being seen that users has limited them to few numbers of the app and spends most of the time there. So the post- app era demands a new trend which can be chatbot. The user usually searches the solutions of their problems in Google, Yahoo, and other search engines but either they do not retrieve concise or relevant information, or they retrieve documents or links to these documents instead of an appropriate answer to their problems. To address such problem the idea of chatbot arises in which user asks in natural language and receives a concise and appropriate answer (Shawar, B.A. and Atwell, E., 2007).

Chatbots can be new updated version of the mobile app or the search engines which will be able to interact with the user in natural language. It is being seen that the limited number of apps in which the users have confined them to are mostly social and messaging apps. So we can surely say it is a positive indication for the development of chatbots. Age of interacting with computers with predetermined commands or clicking on the graphical user interface is long gone, now it is the demand of time the computer starts to take commands in natural language.

With the growth of online-based services like shopping, ordering foods or any official works, it has become necessary to build chatbot to handle large customer base from all over the world at any time.

Recently many developments took place in NLP research so with the large availability of tools for building conversation agent it is the time to transition to taking natural language inputs interface. "The need of conversational agents has become acute with the widespread use of personal machines with the wish to communicate and the desire of

their makers to provide natural language interfaces." (Wilks, 1999). So all the above-mentioned reasons are the factors that played an influential role which motivates us to build a chatbot.

"With nearly 230 million speakers, Bangla is one of the largest spoken languages in the world, but only a very small number of linguistic tools and resources are available for it. For instance, there is no morphological analyzer, POS tagger or syntax parser available for Bangla (Islam, 2012)." As a nation with a history of shedding blood to uphold the honor of Bengali as the national language, we have done nothing for the protection of tools for Bengali Language processing. The development of language resources and its availability is a must for enhancing Language processing capabilities and research in this field (Sarkar, 2007). This very notion motivated us to build first ever Bengali chatbot. With the rise of social messaging apps users among Bengali spoken people, it is the absolute demand of time to build a Bengali chatbot. Also, online based businesses in Bangladesh has a long cherished dream of having an automated system which can take orders in Bengali and replies to users regarding their products in real time 24/7. So all these causes inspired us to take this work at hand.

1.4 Thesis Outline

Chapter One introduces the treatise of human-computer interaction, the need of chatbot and motivation behind this work.

Chapter Two discusses a brief history of chatbot and related works to our systems

Chapter Three explains the methodology, algorithm, flowchart, training and implementation of the prototype.

Chapter Four provides the sample of the experiments done on the systems, comparative studies of Golpo with other chatbot and analysis on it.

CHAPTER II

LITERATURE REVIEW

Early work on chatbots (Weizenbaum 1966) relied on handcrafted templates or heuristic rules to do response generation, which requires huge effort but can only generate limited responses. Recently, researchers begin to develop data driven approaches (Ritter, Cherry, and Dolan 2011; Stent and Bangalore 2014).

Statistical goal-oriented dialogue systems have long been modeled as partially observable Markov decision processes (POMDPs) (Young et al., 2013), and are trained using reinforcement learning based on user feedback. Li et al. (2016) recently applied deep reinforcement learning successfully to train non-goal oriented chatbot type dialogue agents. They show that reinforcement learning allows the agent to model long-term rewards and generate more diverse and coherent responses as compared to supervised learning.

Retrieval based methods select a proper response by matching message response pairs (Hu et al. 2014; Wang et al. 2015; Lu and Li 2013). Retrieval-based methods (Ji, Lu, and Li 2014) retrieve response candidates from a pre-built index, rank the candidates, and select a reply from the top ranked ones.

In related work, we found response selection for retrieval-based chatbots in a single turn scenario, because retrieval-based methods can always return fluent responses (Ji, Lu, and Li 2014) and single turn is the basis of conversation in a chatbot.

Chatbot

A conversational agent which can converse with a human, based on the provided knowledge base and the natural language it was trained on, in any platform eg. mobile, website or desktop application etc is called a chatbot. After Eliza was created, chatbot for long was one of the most sought topics of academic interest among AI researchers. But it was not until 2016, it gained the interest of general mass. With the launch of smartphone based chatbots such as the Apple Siri (Assefi, 2015), Amazon Echo[2], and China's WeChat (Beech, 2014), chatbots turn into one of the hottest trends in technology. Apart from this, some technological giant companies like Facebook Messenger and Skype declared to give full support to the developers for the

[2] A. (n.d.). Amazon Echo - Black. Retrieved April 13, 2017, from https://www.amazon.com/Amazon-Echo-Bluetooth-Speaker-with-WiFi-Alexa/dp/B00X4WHP5E

development of chatbot. Google, the biggest corporation in technology, entered the competition by launching a chatbot application (Allo)[3] powered by its artificial intelligence (AI) and big data.

Human-computer interaction is one of the most difficult challenges in Natural Language Processing (NLP) research. It is a combination of different fields which facilitate communication between users and computers using a natural language depending solely on the language and the available natural language processing techniques (Shawar, 2007). The whole world is entering an era of conversational agents. The era of talking machines is not very far away. In words of Alan Turing, we can say " I propose to consider the question, 'Can machines think?" (Turing, 1950) Much work has been done in information retrieval (IR), machine translation, POS tagging, annotation, and auto-summarization. Although there is quite a large literature on the development of an intelligent machine but still researchers are not successful in making an intelligent machine which can pass Turing Test. Because an intelligent conversational agent is the combination of all the fields of Natural Language Processing (NLP). With the advent of smart personal assistants like Siri, Google Chrome, and Cortana (Meeng, 2011), we may hope for the fulfillment of the dream of Colby. "Before there were computers, we could distinguish persons from non-persons on the basis of an ability to participate in conversations. But now, we have hybrids operating between a person and non-persons with whom we can talk in ordinary language." (Colby 1999a). For achieving this goal AI researchers are working relentlessly to make chatbot which can talk like a human.

The purpose of a chatbot system is to simulate a human conversation; the chatbot architecture integrates a language model and computational algorithms to emulate informal chat communication between a human user and a computer using natural language. Naturally, chatbot can extend daily life, such as help desk tools, automatic telephone answering systems, to aid in education, business, and e-commerce.

Although researchers get success in building chatbot using the retrieval based method but they do not have much success in the generative based method. As Yu (2016) has pointed out the

[3] G. (n.d.). Google Allo. Retrieved April 13, 2017, from https://allo.google.com/

cause of it is not having an appropriated database and the probability of a slightly different answer can lead to a different conversation (Yu, Z., 2016). The main drawback of the generative method is grammatically incorrect and inconsistent sentences.

The present time is the transition period of transforming technology taking commands from predetermined commands to taking inputs from natural language. It is being predicted that chatbot is the future of search engines because it is one of the easiest ways to fetch information from a system. The most important advantage of chatbot based search engine is users can easily search by writing in natural language instead of looking up in a search engine or browse several web pages to collect information.

The chatbot conversation framework falls into two categories: retrieval based and generative based chatbot.

1. Often considered as an easier approach, the retrieval-based model uses a knowledge base of predefined responses and employs a pattern matching algorithm with a heuristic to select an appropriate response. The retrieval based systems do not generate any new text. They can reply on within the domain of their knowledge base.
2. Generative models do not have any knowledge base. So they generate new text in every response. This model relies on the machine translation techniques.

If we compare both the model, we will find advantages and disadvantages in both of them. Since the knowledge base of the retrieval based model is handcrafted by the developer it is not prone to syntactical mistakes. But its disadvantage is it cannot give respond to anything beyond the scope of its knowledge base. On the other hand, generative models are very difficult to train and prone to grammatical mistakes.

The chatbot framework can be again divided into two types based on its domain: closed domain and open domain.

1. The closed domain chatbots are those which can reply to a limited number of subjects. A very good example would be goal based chatbot.
2. An open domain chatbot does not have any knowledge base so it has to generate new sentence for each interaction. Since it has no goal so the users can take the conversation to anywhere. Often unrelated, inconsistent and grammatically incorrect sentences are produced in an open domain modeled chatbot.

So it is very difficult to build a good open domain chatbot which overcomes all the defaults whereas close domain can be easily built if the corpus is available.

As we have discussed the framework let us briefly discuss the internal mechanism of chatbot. There are three important types of artificial intelligence services which are needed to build a chatbot.

1.**Rule-based pattern recognition**:

Mainly any retrieval based chatbot relies on this rule-based pattern recognition.In this model, the rules are the regular expressions.The advantage of a regular expression is that they are flexible and in the case of need new expressions can be created.

2. **Natural language classifier:**

It is used to detect and classify intent of a user command.

3. **Rule-based conversation manager:**

This service can apply rules and generate scripted responses based on the user's intent and data that is associated with the entities, such as location and time.

Therefore, we discuss the definition, the state of art, the classification of the chatbot framework, internal mechanism and classification of artificial intelligence services to build a

chatbot to introduce the background of our work.

CHAPTER III

METHODOLOGY AND IMPLEMENTATION

3.1 Proposed Method

We propose a simple decoder and encoder based conversational model agent that will provide chatbot users with an entity from a Bengali knowledge base (KB) by interactively asking for its attributes.

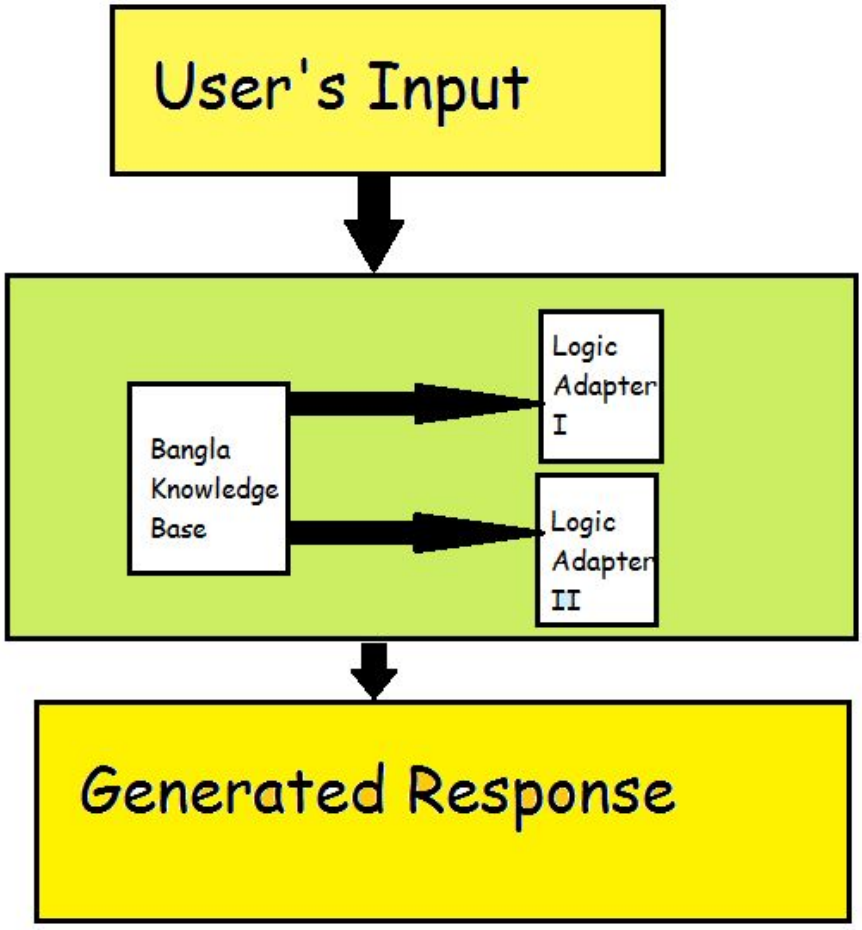


Fig:(3.1(a)) BLOCK DIAGRAM OF Bengali CHATBOT

Most of the works related to conversational agents are done on a retrieval based model. The key to the success of response selection lies in accurately matching input messages with proper responses (Wu, 2016). Our approach for response generation is retrieval based. Retrieval-based model is a model for chatbots which retrieve responses from its knowledge base. It generates a response based on the heuristics, the user's input and the context.

Suppose,

The input to a retrieval-based model is a text t ,

A potential response is r ,

Then,

The output of the model is a confidence score C for the response.

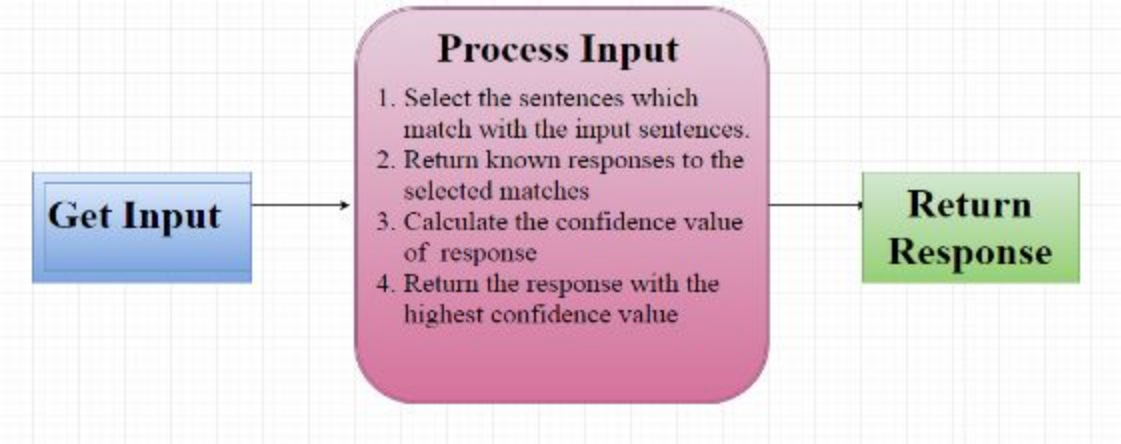
C is a function of ConfidenceValue(t, r).

The r with the highest score C is the response which will be sent to the output adapter.

To find a good response you would calculate the score for multiple responses and choose the one with the highest score. "Selecting a potential response from a set of candidates is an important and challenging task for open-domain human-computer conversation, especially for the retrieval-based human-computer conversation" (Zhou, 2016). Since there is a lot of difficulties in building an open domain chatbot so we want to build a domain specific chatbot. So we are providing it with an initial knowledge base and it can always improve its performance measure by learning from the responses of the users.

The workflow of the chatbot is simple and effective.

1. We will get input from the conversational platform or chat platform.



Figure(3.1(b)) : Flowchart of the Proposed System

2. We will process the received input. The input statement will be processed by an algorithm which will find the best likelihood valued response for the query. The algorithm will select all the known statements that most closely matches the input statement. It will return the known responses to the selected match and a confidence score value based on matching after computation of each of the response. Here the confidence score is the likelihood value of the response. The algorithm will return the response that generated the highest likelihood value for itself.
3. Finally, the response to the input will be returned to the user. For successful completion of user goals, it is also necessary to equip the dialogue policy with real-world knowledge from a database. For constructing this end-to-end system, the following goals can be achieved this by constructing a symbolic query from the current belief states of the agent and retrieving results from the database which match the query.

3.2 Chatterbot

Based on machine learning, ChatterBot[4] is a conversational dialog engine powered by Python which is capable of giving responses based on a knowledge base. We choose this engine

[4] https://github.com/gunthercox/ChatterBot

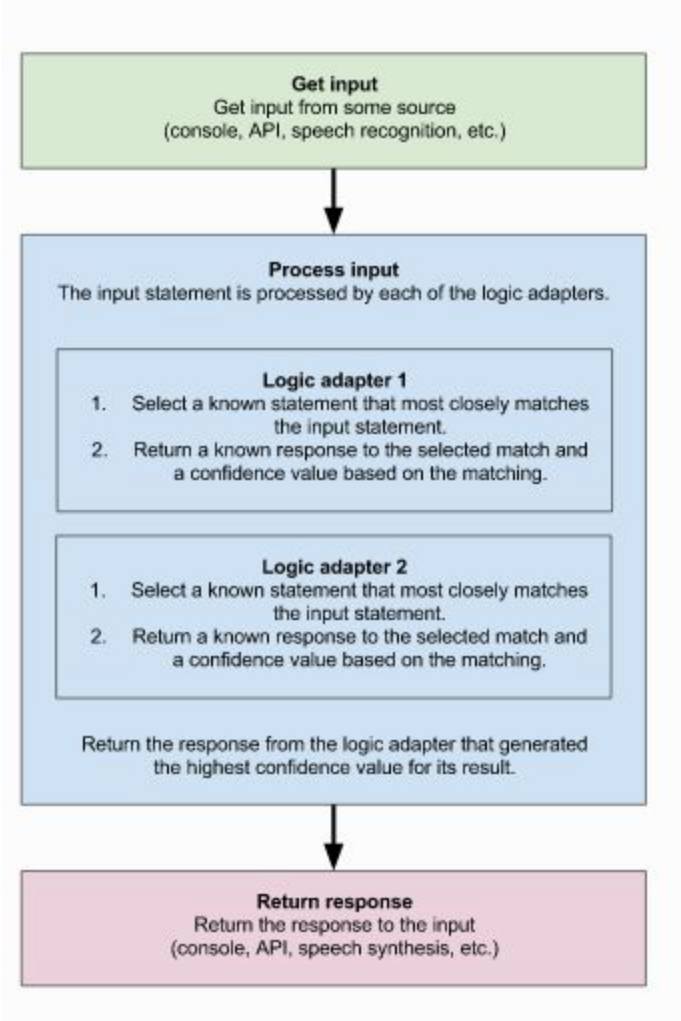
for Golpo because it is language independent. Since Chatterbot has no language dependency in its design, so it is allowed to be trained to speak any language. It is a Python library that makes it easy to generate automated responses to a user's input for the creation of chatbot in any language. To produce different types of responses, ChatterBot applies a selection of machine learning algorithms. This very feature makes it easy for developers to create chatbots and automate conversations with users.

The main class of the chatbot is a connecting point between each of ChatterBot's adapters. In this class, an input statement is returned from the input adapter, processed and stored by the logic and storage adapters, and then passed to the output adapter to be returned to the user.

Additionally, the machine-learning nature of ChatterBot allows an agent instance to improve its own knowledge of possible responses as it interacts with humans and other sources of informative data. 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.[5] The program selects the closest matching response by searching for the closest matching known statement that matches the input, the chatbot then chooses a response from the selection of known responses to that statement.

[5] About ChatterBot. (n.d.). Retrieved April 7, 2017, from https://chatterbot.readthedocs.io/en/stable/chatterbot.html

3.3 Algorithm and Flowchart



Fig(3.3): Process Flow Diagram of ChatterBot[6]

Since Golpo is a retrieval based closed domain chatbot its success lies on the pattern matching algorithm.The algorithm of our system is as follows:

1. Our system takes input from the console or any API, after taking input it sends it to the processing unit.
2. In processing part of the system, there are two logic adapters. The text given by the users

[6] About ChatterBot. (n.d.). Retrieved April 7, 2017, from https://chatterbot.readthedocs.io/en/stable/chatterbot.html

is matched with the existing queries in the database and the sentence which matches with the input is selected. For a selected query, there can be multiple responses in the knowledge base so a confidence score for each of the response for the selected sentence is being calculated. In the second logic adapter, a similar procedure is being carried out.So we get two best responses with highest confidence score from the two adapters. Among the two, the one which got the highest confidence score is sent to the output adapter.

3. The step [1], [2] and [3] continues in a loop until the user exits the console.
4. When the user gives an input, it is stored in the knowledge base as a new query. So with each interaction with the user, the knowledge base learns a new query or response.
5. Our system is able to provide a Bangla corpus from its conversation history.

3.4 Implementation

For the implementation of Bengali chatbot, we have to go through some steps sequentially.

1. The required environmental setup to run the Chatterbot library has been done in our laboratory.
2. After that, we implement a basic English chatbot to get familiar with the system.
3. We prepare a Bengali corpus for building up the knowledge base of the system.
4. There is a particular format to input data in the JSON storage so we have to format the corpus in that format.
5. We write a program which simulates Golpo, a Bengali chatbot.
6. After successful implementation of the system, we do a comparative study with other two chatbots for testing.

Thus we have done the implementation of the Bengali Chatbot.

3.4.1 Environmental Setup

Natural Language Processing (NLP) techniques such as Natural Language Toolkit (NLTK) for Python can be applied to analyze speech, and intelligent responses can be found by designing an engine to provide appropriate human-like responses (Abdul-Kader, 2015). For the setup of Bengali chatbot, we installed Python 3.6 in our Engine. Python is a high-level language which is suitable for scientific research.

Bengali[1][2]

Official Unicode Consortium code chart (PDF)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
U+098x	□	ঁ	ং	ঃ		অ	আ	ই	ঈ	উ	ঊ	ঋ	ঌ			এ
U+099x	ঐ			ও	ঔ	ক	খ	গ	ঘ	ঙ	চ	ছ	জ	ঝ	ঞ	ট
U+09Ax	ঠ	ড	ঢ	ণ	ত	থ	দ	ধ	ন		প	ফ	ব	ভ	ম	য
U+09Bx	র		ল				শ	ষ	স	হ			়	ঽ	া	ি
U+09Cx	ী	ু	ূ	ৃ	ৄ			ে	ৈ			ো	ৌ	্	ৎ	
U+09Dx								ৗ					ড়	ঢ়		য়
U+09Ex	ৠ	ৡ	ৢ	ৣ			০	১	২	৩	৪	৫	৬	৭	৮	৯
U+09Fx	ৰ	ৱ	৲	৳	৴	৵	৶	৷	৸	৹	৺	৻				

Notes

1.^ As of Unicode version 9.0

2.^ Grey areas indicate non-assigned code points

Fig(3.4.1(a)): Official Unicode Consortium Code Chart of Bengali[7]

With availability a large resource of libraries for research purpose, it is the best choice for the natural language processing research. Our chatbot is based on a machine

[7] http://www.unicode.org/charts/PDF/U0980.pdf

learning engine called Chatterbot which is powered by Python. So to run Chatterbot in our machine it is mandatory to install Python.

Fig(3.4.1(b)): Bengali Alphabet

Python 3.6 is recommended for the implementation of Bengali chatbot because any other version below 3.6 of Python causes "Unicode Decode Error". Unicode Decode Error is a runtime error caused by non-English language with a large number of letters in the alphabet. The Unicode range of Bengali is 0980–09FF. It has 11 vowels and 40 consonants. Unlike English, it has consonant conjuncts, modifier, and other graphemes. So Bengali cannot be dealt with ASCII decoding. For easy installation of Chatterbot, it is recommended to install Anaconda for the setup. Anaconda[8] is an open source data science platform powered by Python. Only Python 3.6 and Anaconda 3 supports taking the input in Bengali from a database. So it is advisable to use Python 3.6 and Anaconda 3 for this purpose.

There is required software to run Chatterbot in any engine. We installed chatterbot-corpus for the implementation of the English Chatbot. It is notable to mention that after implementation of Golpo we add the Bangla corpus to the chatterbot corpus. So due to our contribution, anyone installing Chatterbot corpus will also get a sample Bengali corpus[9] made by

[8] https://www.continuum.io/downloads

[9] https://github.com/gunthercox/chatterbot-corpus/tree/master/chatterbot_corpus/data/bangla

us. For storing the knowledge base, we install JSON database.

For the implementation of the English chatbot, we installed Natural Language Toolkit

Fig:(3.4.1(c)): Required Tools for Environmental Setup of Golpo

(NLTK) for Python. These were the requirements for the setup of chatterbot. After all required tools are installed, we installed Chatterbot in our Engine. We use Pycharm Community Edition as an IDE for writing, compiling and running program.

Thus, the environmental setup for Golpo has been done.

3.4.2 Training

ChatterBot includes tools that help simplify the process of training a chat bot instance. ChatterBot's training process involves loading example dialog into the chat bot's database. This either creates or builds upon the graph data structure that represents the sets of known statements and responses. When a chat bot trainer is provided with a data set, it creates the necessary entries in the chat bot's knowledge graph so that the statement inputs and responses are correctly represented.

Several training classes come built-in with ChatterBot. These utilities range from allowing you to update the chat bot's database knowledge graph based on a list of statements representing a conversation, to tools that allow you to train your bot based on a corpus of preloaded training data.

The training of Golpo can be done in two processes.

1. **Training via list Data:**

   This training process allows a chatbot to be trained using a list of strings where the list represents a conversation.In this case, the order of each response is based on its placement in a given conversation or the list of string.

1. Import Chatterbot and Trainer Class Library
2. Set the trainer as List Data Trainer
3. Provide the list of strings which you want to provide for training
4. Train the chatbot
5. Give input and get response

Figure(3.4.2(a)): Pseudocode of an Instance showing how to use the List Trainer Class

1. List Trainer Class is a class with a method called Train.This class allows a chatbot to be trained using a list of strings where the list represents a conversation.
2. In Train method, initialize a data structure to store the history of conversation.
3. Store each statement from the provided list to the initialized data structure

Figure (3.4.2(b)): Pseudocode for the List Trainer Class

2. Training via Corpus Data:

ChatterBot comes with a corpus data and utility module that makes it easy to quickly train your bot to communicate. To do so, simply specify the corpus data modules you want to use. This training class allows the chatbot to be trained using data from the dialog corpus.

```
from chatterbot.trainers import ChatterBotCorpusTrainer

chatterbot = ChatBot("Training Example")
chatterbot.set_trainer(ChatterBotCorpusTrainer)

chatterbot.train(
    "chatterbot.corpus.bangla"
)
```

Figure (3.4.2(c)): Code Snippet of the Bengali Chatterbot's Training Class

For our implementation, we used the Corpus Trainer Class of Chatterbot. At first, we have to create Bangla corpus in the data folder of the Chatterbot in the

predefined format in JSON. So from the library, we set the trainer as training with the Bangla Corpus.We are providing the pseudo code of corpus trainer class based on the code by Gunther Cox for a better understanding of the trainer class.[10]

1. Corpus Trainer Class is a class with a method called Train.This class allows the chatbot to be trained using data from the ChatterBot dialog corpus.

2. Import the corpus of the language mentioned in the command for the chatterbot-corpus library
3. In Train method, initialize a data structure to store the history of conversation.
4. Check whether length of the corpus is larger than the capacity of the storage. If it is larger, then return out of space error otherwise start training
5. Store each statement from the provided list to the initialized data structure.

Figure (3.4.2(d)): Pseudo code for corpus trainer class of Chatterbot

3.4.3 Storage Adapters

ChatterBot comes with built-in adapter classes that allow it to connect to different types of databases. For our implementation, we will be using the Json File Storage Adapter which is a simple storage adapter that stores data in a JSON formatted file on the hard disk. This functionality makes this storage adapter very good for testing and debugging.

[10] https://github.com/gunthercox/ChatterBot

```
bot = ChatBot(
    'Orin',
    storage_adapter='chatterbot.storage.JsonFileStorageAdapter',
    database='./database.json'
)
```

Figure (3.4.3): Code snippet of initialization of the storage adapter of our system

We will select the Json File Storage Adapter by specifying it in our chat bot's constructor.

The database parameter is used to specify the path to the database that the chat bot will use. The database.json file will be created automatically if it does not already exist.

3.4.4 Input Adapters

ChatterBot's input adapters are designed to allow a chat bot to have a versatile method of receiving or retrieving input from a given source. It is required to add in parameters to specify the input and output terminal adapter. The input terminal adapter simply reads the user's input from the terminal.

The Chatterbot's input adapter class is an abstract class that represents the interface that all input adapters should implement. After getting input, the main job is the classify the text as a known or an unknown statement and pass it to the logic adapter after labeling the sentence as "known" or "unknown". The goal of an input adapter is to get input from some source, and then to convert it into a format that ChatterBot can understand. This format is the Statement object found in ChatterBot's conversation module.

We used the variable input adapter for the implementation of Golpo. Variable input type adapter allows the chatbot to accept a number of different input types using the same adapter. This adapter accepts strings, dictionaries, and statements.

3.4.5 Output Adapters

The output adapter allows the chatbot to return a response in as a Statement object.It is a generic class that can be overridden by a subclass to provide extended functionality, such as delivering a response to an API endpoint.

Since our system is a text-based system we chose the “Text” format for our chatbot.

```
chatbot = ChatBot(
    "Orin",
    output_adapter="chatterbot.output.OutputAdapter",
    output_format="text"
)
```

Figure(3.4.5): Code Snippet of initialization of output Adapter of Golpo

3.4.6 Logic Adapters

Logic adapters determine the logic for how ChatterBot selects responses to a given input statement. The logic adapter that your bot uses can be specified by setting the logic_adapters parameter to the import path of the logic adapter you want to use.

It is possible to enter any number of logic adapters for your bot to use. If multiple adapters are used, then the bot will return the response with the highest calculated confidence value. If multiple adapters return the same confidence, then the adapter that is entered into the list first will take priority.

The logic_adapters parameter is a list of logic adapters. In ChatterBot, a logic adapter is a class that takes an input statement and returns a response to that statement.

1. **BestMatch logic adapter is a logic adapter that returns a response based on known responses to the closest matches to the input statement.**
2. **Import the unicode literals and the logic adapter library**
3. **In Get method, it takes a statement string and a list of statement strings and returns the closest matching statement from the list.**
4. **If no statements have known responses then the Get method chooses a random response to return. For random response, the Get method sets the confidence score as zero.**
5. **For known statement(s), the Get method calculated the confidence score by doing "Jaccard Similarity" comparison and return the one with highest confidence score.**

Fig(3.4.6): Pseudocode for the Best Match Logic Adapter

We employ Best Match Adapter for our chatbot. It is a logic adapter that returns a response based on known responses to the closest matches to the input statement. The Best Match logic adapter selects a response based on the best known match to a given statement. Once it finds the closest match to the input statement, it uses another function to select one of the known responses to that statement.

The best match adapter uses Jaccard Similarity function to compare the input statement to known statements. Jaccard Similarity compared two sentences based on Jaccard Index.The Jaccard index is a ratio of numerator and denominator or in other words, it is a fraction. In the numerator, we count the number of items that are shared between the sets. In the denominator, we count the total number of items across both sets. Let's say we define sentences to be equivalent if 50% or more of their tokens are equivalent.

Here are two sample sentences:

The young cat is hungry. The cat is very hungry.

When we parse these sentences to remove stop words, we end up with the following two sets:

{young, cat, hungry} {cat, very, hungry}

In our example above, our intersection is {cat, hungry}, which has a count of two. The union of the sets is {young, cat, very, hungry}, which has a count of four. Therefore, our Jaccard similarity index is two divided by four, or 50%. Given our threshold above, we would consider this to be a match.

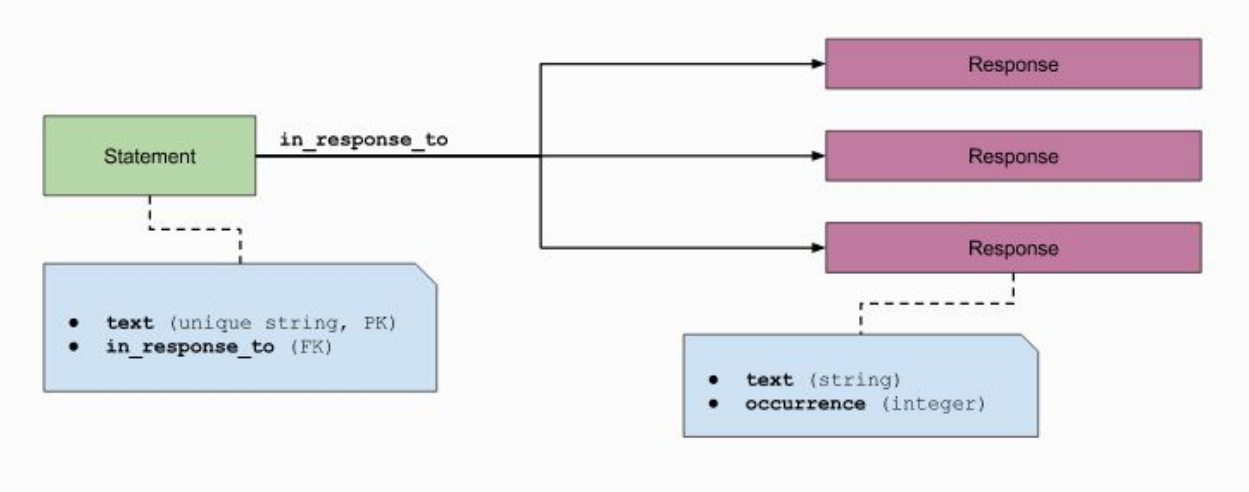
3.4.7 Response Selection Methods

Response selection methods determine which response should be used in the event that multiple responses are generated within a logic adapter. ChatterBot uses Statement objects to hold information about things that can be said. An important part of how a chat bot selects a response is based on its ability to compare two statements to each other. This module contains various text comparison algorithms designed to compare one statement to another.

We use the get first response method for the selection of a response. This method takes the input statement and selects the statement in the knowledge base which closely matches the input to the chatbot from a list of statement options to choose a response from.

3.4.8 Statement-Response Relationship

ChatterBot stores knowledge of conversations as statements. Each statement can have any number possible responses.



Figure(3.4.8(a)) : The Relationship between Statement and responses

Each Statement object has an in_response_to reference which links the statement to a number of other statements that it has been learned to be in response to.

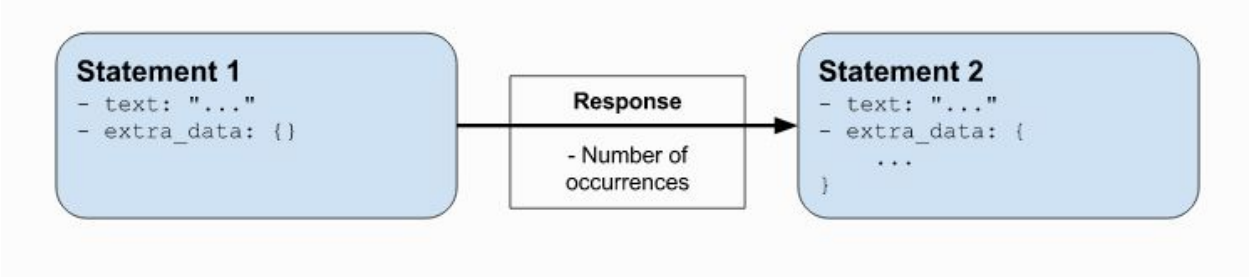


Figure (3.4.8(b)): Mechanism of the reference to all parent statements of the current Statement.

The in_response_to attribute is essentially a reference to all parent statements of the current

statement.

The Response object's occurrence attribute indicates the number of times that the statement has been given as a response. This makes it possible for the chatbot to determine if a particular response is more commonly used than another.

3.4.9 Dataset Format

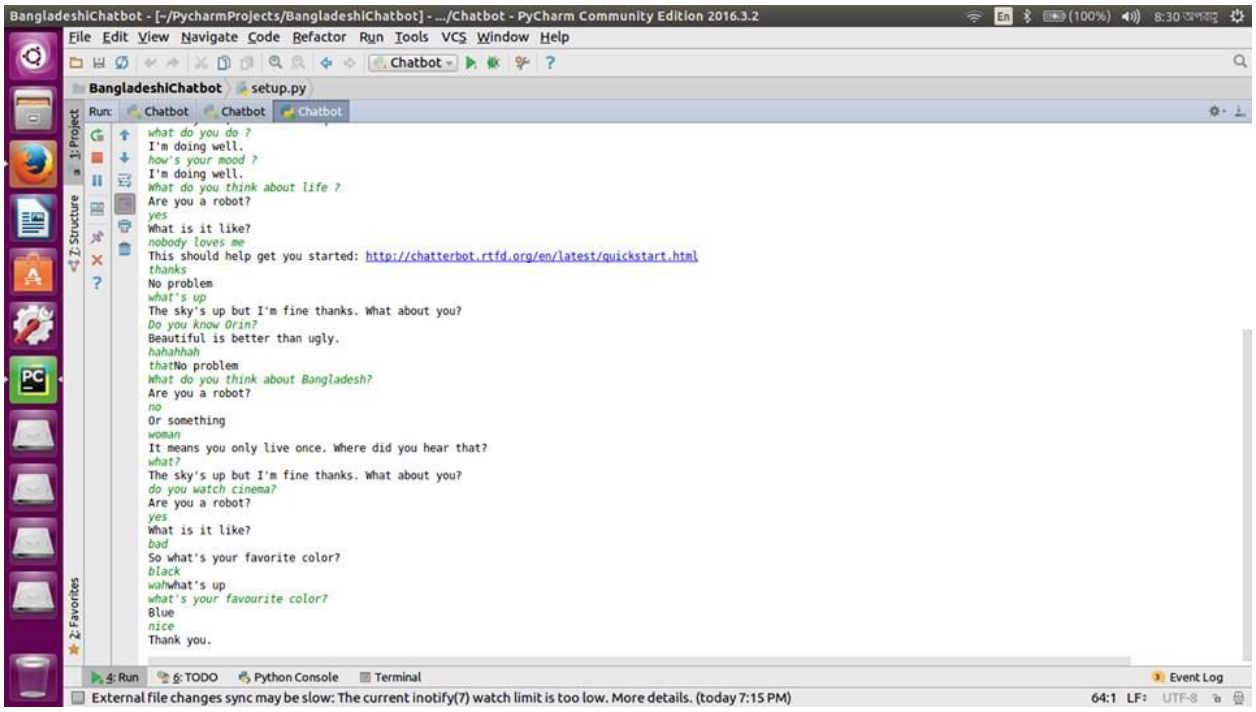
```
{
    "প্রোফাইল": [
        [
            "তোমার আগ্রহগুলো কি কি",
            "আমি সব ধরনের জিনিস আগ্রহী। আমরা কিছু বিষয়ে কথা বলতে পারবেন না। আমার প্রিয় বিষয় রোবট এবং কম্পিউটার, প্রাকৃতিক ভাষা প্রক্রিয়া
        ],
        [
            "আপনার ফোন নাম্বার কত",
            "নেই"
        ],
        [
            "আপনি কেন খাওয়া যাবে না",
            "আমি বিদ্যুৎ গ্রাস হবে"
        ],
        [
            "আপনার অবস্থান কী",
            "সবখানে"
        ],
        [
            "তোমার কি কোন ভাই আছে",
            "আমি কোন ভাই নেই। কিন্তু আমি ক্লোনস অনেক আছে।"
        ],

        [
            "আপনার বয়স কত",
            "আমি এখনও তরুণ "
        ],
    ]
}
```

Figure (3.4.9): Format for Creating the Corpus for JSON Storage

This format must be followed for the manual creation of any corpus. It is to be noted that the generated corpus from Golpo will be in the same format.

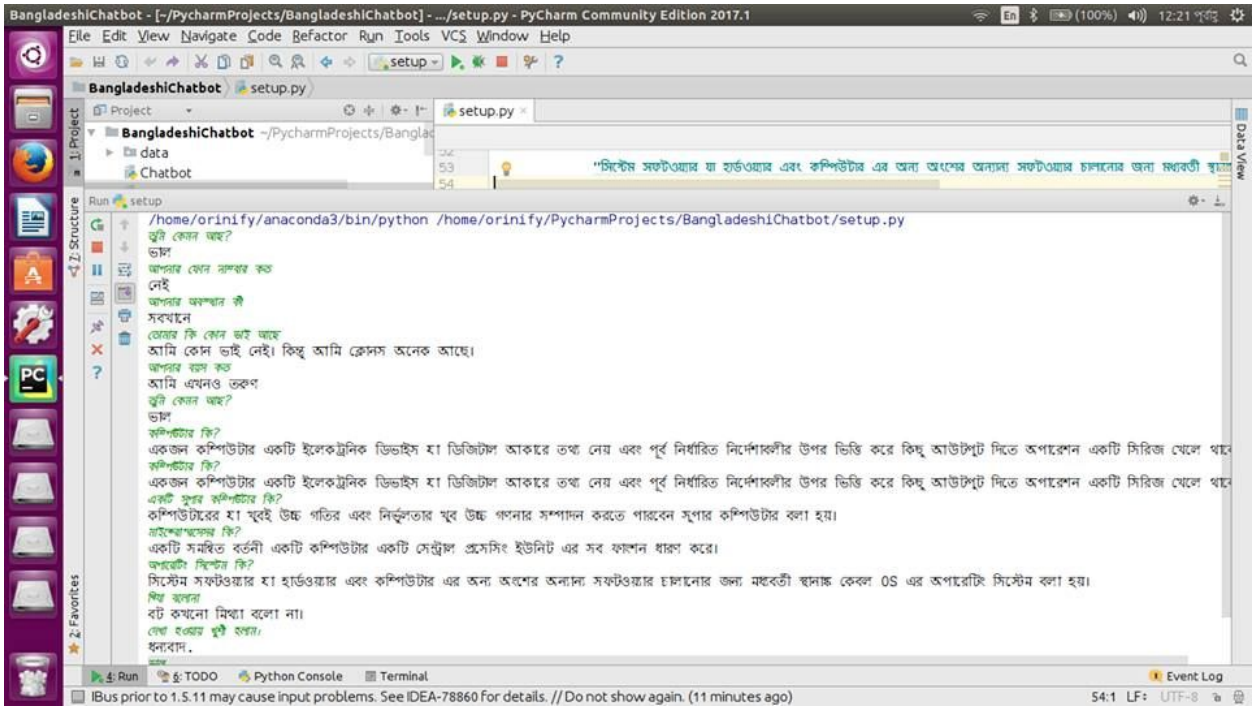
3.4.10 Implementation of Prototype



Figure(3.4.10(a)) : Implementation of the English Chatbot

To get familiar with the system, we implemented the English corpus and tried different parameter for input adapter, output adapter, and logic adapter. After experimenting, we select the best parameter for our work. The corpus used for the implementation of the English chatbot is the sample corpus in Chatterbot corpus library.We successfully implement the English chatbot. Our chatbot produces similar responses in Bangla as its English counterpart.

We implement Golpo based on a manually created Bangla corpus to converse in basic Bangla. For input adapter, we select the variable input adapter and for output adapter, we select the text form. Best match logic adapter has been selected for the response generation logic. The comparison algorithm used inside the best match logic adapter is the Jaccard Similarity Comparison algorithm.



Figure(3.4.10(b)) : Implementation of Golpo

The main focus of our work is to generate sentences free from grammatical mistakes, spelling mistakes and consistent. Golpo achieves the goal of producing syntactically correct Bengali responses. Since the responses are as good as its knowledge base so a lot of work has to be done to enhance the knowledge base. Topic-wise data can be fed to Golpo for the enhancement of its knowledge base. But during building the knowledge base, the developer must provide a knowledge base free from errors.

CHAPTER IV

RESULTS AND ANALYSIS

The difficulty of evaluation is intrinsic as each conversation is interactive, and the same conversation will not occur more than once; one slightly different answer will lead to a completely different conversation; moreover, there is no clear sense of when such a conversation is "complete" (Yu, 2016). So for the evaluation, we decided to compare our system with previous existing chatbots.

4.1 Comparison between English Chatbot and Bengali Chatbot

Since this is the first chatbot in Bengali, so there is no other Bengali chatbot to compare with our work. Our work is at present the state of the art for the Bengali chatbot. So we compare Golpo with two English chatbots which are Neural Conversational Machine (NCM) and Cleverbot.

Neural Conversation Machine is a chatbot which can be trained end-to-end and this model chats by predicting the next sentence given the previous sentences or sentence with a few handcrafted rules. (Vinyals, 2015) So NCM is a generative based conversational agent whereas our work is a retrieval based system. We input some random questions to the NCM and our system for the comparative study between the two.

User	NCM	Golpo	Comment
what is two plus two? দুই যোগ দুই কত?	Uh, well, we got four.	চার	Quite Similar but the response of NCM is quite natural while Golpo replied in a straight way.
what is the color of a yellow car? একটি হলুদ গাড়ির রঙ কি?	yellow	হলুদ	Same
what is the meaning of life? বেঁচে থাকার উদ্দেশ্য কী?	i mean, what is life?	ভাল কাজ করার জন্য।	Golpo gives a better reply than NCM. The response of NCM is not exact but prevalent.

Table(1): The comparative study between NCM and Golpo

Cleverbot is very similar to our system. Cleverbot's responses are not pre-programmed. Instead, it learns from human input. Humans type into the box below the Cleverbot logo and the system finds all keywords or an exact phrase matching the input. After searching through its conversation history, it responds to the input by finding how a human responded to that input when it was asked, in part or in full, by Cleverbot.

User	Cleverbot	Golpo	Comment
What are your interests? তোমার আগ্রহ-গুলো কি কি	I like reading and writing.	আমি সব ধরনের জিনিষে আগ্রহী। আমার প্রিয় বিষয় রোবট এবং কম্পিউটার, প্রাকৃতিক ভাষা প্রক্রিয়াকরণ ।	Both answers are relevant and correct. But Golpo gave a more explanatory and natural answer.
What is your number? আপনার ফোন নাম্বার কত	My name is AnneMeansGrace. What is your name?	নেই	While Golpo's reply is precise and relevant one Cleverbot replied an irrelevant answer and asked an irrelevant question.
What is a computer? কম্পিউটার কি**?**	What do you mean?	কম্পিউটার একটি ইলেকট্রনিক্স ডিভাইস যা ডিজিটাল আকারে তথ্য নেয় এবং পূর্ব নির্ধারিত নির্দেশনাবলীর উপর ভিত্তি করে কিছু আউটপুট দিয়ে থাকে।	Golpo outperforms Cleverbot by providing a definition of the query whereas Cleverbot instead of replying asked a question.

Table(2): The comparative study between Cleverbot and Golpo

Similarly, an untrained instance[11] of our work based on Chatterbot starts off with no knowledge of how to communicate. Each time a user enters a statement, the database stores the input that they entered and the text that the statement was in response to. As Golpo 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 chooses a response from the selection of known responses to that statement. So we evaluate our chatbot by comparing with a related work like Cleverbot.

4.2 Analysis

Since we do not find a suitable database for this purpose we manually prepared and annotated conversation corpus in Bengali. Our work addresses the problem of developing a Bengali chatbot in spite of required language processing tools like Parts Speech Tagger, Tokenizer etc. We solve the problem of lack of required tools by selecting a language independent platform and choosing a retrieval based model to serve the purpose. In the case of the evaluation, we face the same problem since this work is the pioneering work in the Bengali Conversational Agent there is no benchmark for the evaluation of the chatbot in Bengali.So we are left with an option of comparing it with any English chatbot. Therefore, we compare Golpo with two popular chatbots which are Neural Conversational Model (NCM) and the Cleverbot.

To ensure a fair comparison, the questions asked in English is an exact translation of the questions asked in Bengali. NCM is a neural based open domain generative based chatbot where ours is a retrieval based closed domain one. In the experiments, our chatbot gives the similar response as the NCM. Cleverbot is a chatbot hosted on a website who learns from the user and answers based on the conversation history which is quite similar to our work. It is interesting to observe that our system outwitted Cleverbot in many cases. We examine Golpo by inputting unknown sentences as a test case and find Golpo to produce random answers to the questions. But it stores the reply given by the user to the unknown sentences and later on replies the same

[11] https://chatterbot.readthedocs.io/en/stable/#term-untrained-instance

answer to another user in another instance. Golpo is able to reply in real time like others. Since it can take input in Bengali and can give a response in Bengali so we can say that the pattern matching algorithm is functioning well. Our chatbot replies in syntactically correct Bengali and it is free from spelling mistakes and any sort of grammatical mistakes. It makes some punctuation mistake which can be improved in future.

From the samples, we can see our Bengali chatbot Golpo gives a similar reply like Neural Conversational Machine (NCM) whereas in comparison with a related work Cleverbot our Golpo has outwitted it in most of the instance.Amongst the many limitations, the lack of a coherent personality makes it difficult for our system to pass the Turing test (Turing, 1950).

Our work provides us with a conversation corpus in Bengali. This generation of the corpus has many advantages. Corpus is considered as a basic resource for language analysis and research for many foreign languages. This reflects both ideological and technological change in the area of language research. This change is probably caused due to the introduction of computer and corpus in linguistic research which, as a result, have paved out many new applications of language (and linguistics) in the fields of communication and information exchange. The use of Bengali language corpus for various technological developments as well as for various linguistic studies in the Bengali language can open up many new avenues for us. This corpus can be useful for producing many sophisticated automatic tools and systems, besides being good resources for language description and theory making.

CHAPTER IV

CONCLUSION AND FUTURE WORK

5.1 Conclusion

This research is a pioneering work in the field of dialogue system in Bengali. The main challenge of this work is to create a chatbot based on the accurate knowledge base. Due to lack of large dataset, we implemented a retrieval based closed domain chatbot which will converse with the user based on the pattern matching algorithm and will improve its performance measure by learning from the interaction. Our work will provide a Bengali conversation corpus which will help in the development of tools for Bengali Language Processing research.

5.2 Future Work

For future work, one aspect to enhance the knowledge base of Golpo is to host it on a crowdsourcing platform.[12] Like Google Translate Bengali, Golpo will be able to learn from the interactions with users. The more the number of interactions, the more will be the percentage of relevant replies to a query can be provided by Golpo.

For future works, we can train a chatbot with a neural network model provided with the corpus we get from this project. Also, we can try a crowd-sourced model to enrich the database of the chatbot by integrating this chatbot in a website.

The context can be incorporated in future work. To produce sensible responses, systems may need to incorporate both linguistic context and physical context. In long conversations, people keep track of what has been said and what information has been exchanged. One may also need to incorporate other kinds of contextual data such as date/time, location, or information

[12] Crowdsourcing is the process of procuring required services, ideas, or contents by soliciting contributions from a large group of people and especially from the online community rather than from traditional employees or suppliers

about a user. Since it requires a large collection of conversation corpus to make a generative open domain system to incorporate context. As we do not have that data to make it generative so this can be done in future.

We can optimize the automated system by making the chatbot a voice-enabled system, to reply in pictorial representation for better understanding for people with low literacy.

REFERENCES

(Abdul-Kader, 2015) Abdul-Kader, S. A., & Woods, J. (2015). Survey on chatbot design techniques in speech conversation systems. Int. J. Adv. Comput. Sci. Appl.(IJACSA), 6(7).

(Beech, 2014) Beech, H. (2014). What's All The Fuss About Whats App? China's WeChat Is a Worthy Rival. Time. com, 2, 1.

(Colby 1999a) Colby, K. 1999a. Comments on the human-computer conversation. In Wilks, Y. (ed.), Machine Conversations. Kluwer, Boston/Drdrecht/London, pp. 5-8.

(Hu et al. 2014) Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In Advances in Neural Information Processing Systems, pages 2042–2050.

(Islam, 2012) Islam, Z., Mehler, A., Rahman, M. R., & Text technology, A. G. (2012, November). Text Readability Classification of Textbooks of a Low-Resource Language. In PACLIC (pp. 545-553)

(Ji, Lu and Li 2014) Zongcheng Ji, Zhengdong Lu, and Hang Li. 2014. An information retrieval approach to short text conversation. arXiv preprint arXiv:1408.6988

(Li et al. ,2016) Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A persona based neural conversation model. arXiv preprint arXiv:1603.06155.

(M. J. Pereira, 2013) M. J. Pereira, and L. Coheur, "Just. Chat-a platform for processing information to be used in chatbots," 2013.

(Meeng, 2011) Meeng, M., & Knobbe, A. (2011). Flexible enrichment with Cortana–software demo. In Proceedings of BeneLearn (pp. 117-119).

(Ritter, Cherry, and Dolan 2011) Ritter, A., Cherry, C., & Dolan, B. (2010, June). Unsupervised modeling of twitter conversations. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (pp. 172-180). Association for Computational Linguistics.

(Saenz, 2010) Saenz, A. (2010). Cleverbot chat engine is learning from the internet to talk like a human. Singularity Hub.

(Shang, Lu, and Li 2015) Shang, L.; Lu, Z.; and Li, H. 2015. Neural responding machine for short-text conversation. In ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers, 1577–1586.

(Shawar, 2007) Shawar, B. A., & Atwell, E. (2007, April). Different measurements metrics to evaluate a chatbot system. In Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies (pp. 89-96). Association for Computational Linguistics.

(Shawar, 2007) Shawar, B.A. and Atwell, E. (2007) Chatbots: are they really useful? Journal of Computational Linguistics and Language Technology, Vol. 22, No. 1, pp.29-49.

(Sarkar, 2007) Sarkar, A. I., Pavel, D. S. H., & Khan, M. (2007). Automatic Bangla corpus creation. BRAC University.

(Stent and Bangalore, 2014) Stent, A., & Bangalore, S. (Eds.). (2014). Natural language generation in interactive systems. Cambridge University Press.

(Turing, 1950) Turing, A. M. (1950). Computing machinery and intelligence. Mind, 59(236), 433-460.

(Vinyals, 2015) Vinyals, O., & Le, Q. (2015). A neural conversational model. arXiv preprint arXiv:1506.05869.

(Wang et al. 2015) Hao Wang, Zhengdong Lu, Hang Li, and Enhong Chen. 2013. A dataset for research on short text conversations. In EMNLP, pages 935–945.

(Weizenbaum, 1966) Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. Communications of the ACM, 9(1), 36-45.

(Wilks, 1999) Wilks, Y. (1999). Preface. In Wilks, Y., editor, Machine Conversations, pages vii–x. Kluwer, Boston/-Dordrecht/London.

(Wu, 2016) Wu, Y., Wu, W., Li, Z., & Zhou, M. (2016). Response Selection with Topic Clues for Retrieval-based Chatbots. arXiv preprint arXiv:1605.00090.

(Young et al., 2013) Steve Young, Milica Gasiˇ c, Simon ´ Keizer, Franc¸ois Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A

practical framework for pomdp-based spoken dialogue management. Computer Speech & Language, 24(2):150–174.

(Yu, Z., 2016) Yu, Z., Xu, Z., Black, A., & Rudnicky, A. (2016). Chatbot evaluation and database expansion via crowdsourcing. In Proc. of the chatbot workshop of LREC.

(Zhou, 2016) Zhou, X., Dong, D., Wu, H., Zhao, S., Yan, R., Yu, D., ... & Tian, H. (2016). Multi-view response selection for human-computer conversation. EMNLP'16.