Course Title: Neural Networks

Course Code: CSE425

Project Title: Implementation of Bangla ChatBot Using Transformer-Based Models

Group No-03

Name	Student ID	Section
Farhan Faruk	20301137	03
Aunon Halder	20301133	03
H.M Sarwer Alam	20301224	03
Saif Mokarrom	20301121	03
Shudeb Ghosh Barno	20301222	03

1. Introduction

The absolute goal of Artificial Intelligence research is to build a machine that 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 at the 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 that 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 can perform actions based on the conversations (goal-oriented chatbots, for example, pizza ordering chatbot). It is popularly known as a chatbot, 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 or Simsimi, automated tutorials, and online 1 assistants as chatbots. The use cases of chatbots include customer care representatives, sales agents, FAQ answerers, etc. With the rise of smartphones and other high computational devices, chatbots have become one of the hot topics of Natural Language Processing (NLP) research. This is to be noted that all research was based on only one language which is English. Although some works have been done in Chinese and Spanish languages there are very few. Other languages were left out because of the lack of quality data corpus and natural language 1 http://www.cleverbot.com/ Golpo: Implementation of a Bangla Chatbot 11 processing tools. Bengali is the fourth most spoken language in the world. It is the mother tongue of the people of Bangladesh and also the language of some parts of India such as Calcutta. But it is a matter of great regret that it is considered a 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 in the 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 the unavailability of a 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 that will able to interact in Bengali as well as build a corpus in Bengali. We name it "Golpo" which means "Story" in Bengali. As the 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 the lack of quality corpus in Bengali, the job becomes even more complicated. So for any researcher 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 did not find a suitable database for this purpose we manually prepared and annotated the conversation corpus. In this work, for the implementation, we create an appropriate Bengali corpus from scratch and design it in such a way that it can provide us with a generated Bengali corpus. Since it is the first attempt we are using the retrieval-based system based on a pattern-matching mechanism between the inputs and the hand-crafted rules predetermined in the Golpo: Implementation of a Bangla Chatbot 12 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 twofold. (1) We start the research of a Bengali chatbot or conversational agent (2) Our system can provide a Bengali corpus that can be used for other Bengali Language Processing research in the future. We evaluate the chatbot based on user satisfaction. Shawar, B. A., & Atwell, E. (2007, April) advised that "Evaluation should be adapted to the application and 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 can 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.

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 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.

3. Motivation

The last decade was the decade where a technological revolution took place. With the increase in the usage of smartphones, the number of social network users and mobile app users has increased manifold. Now the question is “What’s next?”. From recent research on the pattern of consumer behavior related to smartphones, it is being seen that users have limited themselves to a few numbers of apps and spend most of their time there. So the post-app era demands a new trend which can be chatbots. 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 a problem the idea of a chatbot arises in which the user asks in natural language and receives a concise and appropriate answer (Shawar, B.A. and Atwell, E., 2007). Chatbots can be new updated versions of the mobile app or search engines that will be able to interact with the user in natural language. It is being seen that the limited number of apps to which the users have confined them are mostly social and messaging apps. So we can surely say it is a positive indication for the development of chatbots. The 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 food, or any official work, it has become necessary to build chatbots to handle large customer bases from all over the world at any time. Recently many developments have taken place in NLP research so with the large availability of tools for building conversation agents it is time to transition to taking natural language inputs interface. "The need for conversational agents has become acute with the widespread use of personal machines with the wish to communicate and the desire of Golpo: Implementation of a Bangla Chatbot 14 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, no morphological analyzer, POS tagger, or syntax parser is 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 their availability is a must for enhancing Language processing capabilities and research in this field (Sarkar, 2007). This very notion motivated us to build the first-ever Bengali chatbot. With the rise of social messaging app users among Bengali-spoken people, it is the absolute demand of time to build a Bengali chatbot. Also, online-based businesses in Bangladesh have a long-cherished dream of having an automated system that can take orders in Bengali and reply to users regarding their products in real-time 24/7. So all these causes inspired us to take this work at hand.

4. 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 have begun 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.

5. Chatbot

A conversational agent that 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, 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, that it gained the interest of the general mass. With the launch of smartphone-based chatbots such as Apple Siri (Assefi, 2015), Amazon Echo, and China's WeChat (Beech, 2014), chatbots turn into 2 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/B00X4WHP5 E Golpo: Implementation of a Bangla Chatbot 16 development of chatbot. Google, the biggest corporation in technology, entered the competition by launching a chatbot application (Allo) powered by its artificial intelligence (AI) and big data. 3 Human-computer interaction is one of the most difficult challenges in Natural Language Processing (NLP) research. It is a combination of different fields that 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 the 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 still researchers are not successful in making an intelligent machine that can pass the Turing Test. Because an intelligent conversational agent is a 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 our dream of Colby. “Before there were computers, we could distinguish persons from non-persons based on 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). To achieve this goal AI researchers are working relentlessly to make a chatbot that 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, chatbots can extend daily life, such as help desk tools, and automatic telephone answering systems, to aid in education, business, and e-commerce. Although researchers have success in building chatbots using the retrieval-based method 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/. Because it does not have an appropriate 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 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 engines is users can easily search by writing in natural language instead of looking up in a search engine or browsing several web pages to collect information. The chatbot conversation framework falls into two categories: retrieval-based and generative-based chatbot. 1. Often considered 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 rely 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 machine translation techniques. If we compare both 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 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 a new sentence for each interaction. Since it has no goal the users can take the conversation 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 that overcomes all the defaults whereas a closed 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. Three important types of artificial intelligence services 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 the 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 the art, the classification of the chatbot framework, the internal mechanism, and the classification of artificial intelligence services to build a chatbot to introduce the background of our work.

6. METHODOLOGY AND IMPLEMENTATION

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.

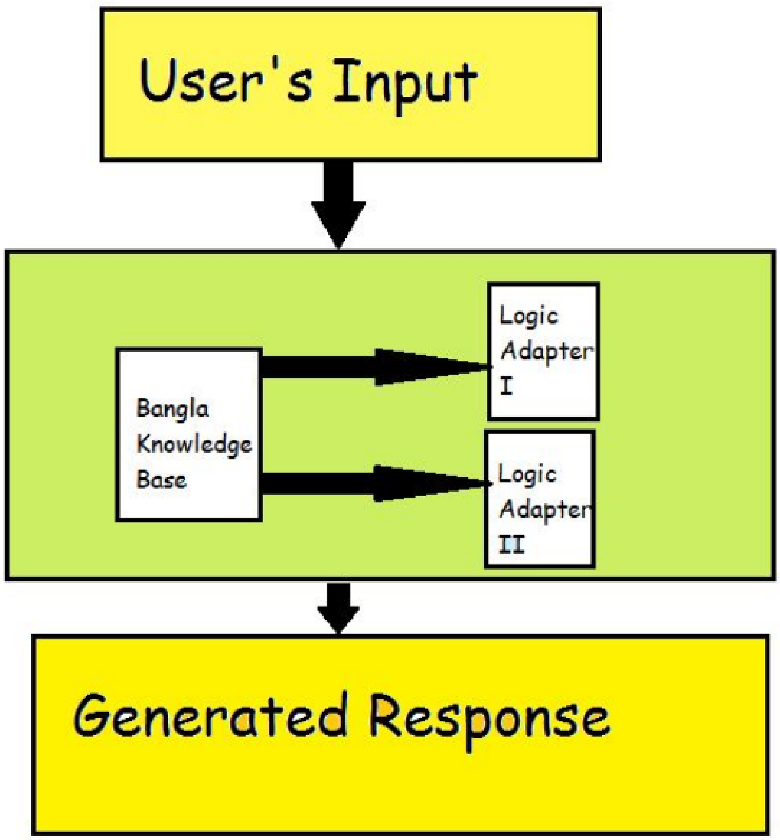

Fig: BLOCK DIAGRAM OF Bengali CHATBOT

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 that retrieve responses from their 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, 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 that 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 are a lot of difficulties in building an open-domain chatbot 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.

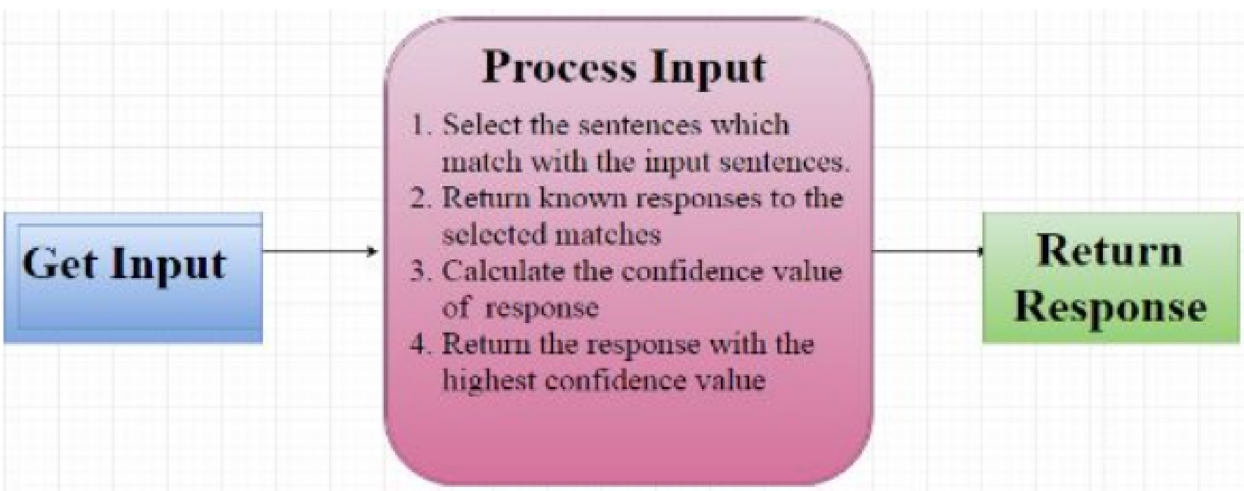

Figure: Flowchart of the Proposed System

2. We will process the received input. The input statement will be processed by an algorithm that will find the best likelihood-valued response for the query. The algorithm will select all the known statements that most closely match the input statement. It will return the known responses to the selected match and a confidence score value based on matching after the computation of each of the responses. 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 the 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 by constructing a symbolic query from the current belief states of the agent and retrieving results from the database that match the query. 3.2 Chatterbot Based on machine learning, ChatterBot is a conversational dialog engine powered by 4 Python which is capable of giving responses based on a knowledge base. We chose this engine 4 https://github.com/gunthercox/ It is language independent. Since Chatterbot has no language dependency in its design, 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 a 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 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 to and the accuracy of each response concerning the input statement increase. The program selects the closest matching response by searching for the closest 5 matching known statement that matches the input, the chatbot then chooses a response from the selection of known responses to that statement.

7. Algorithm’s Flowchart

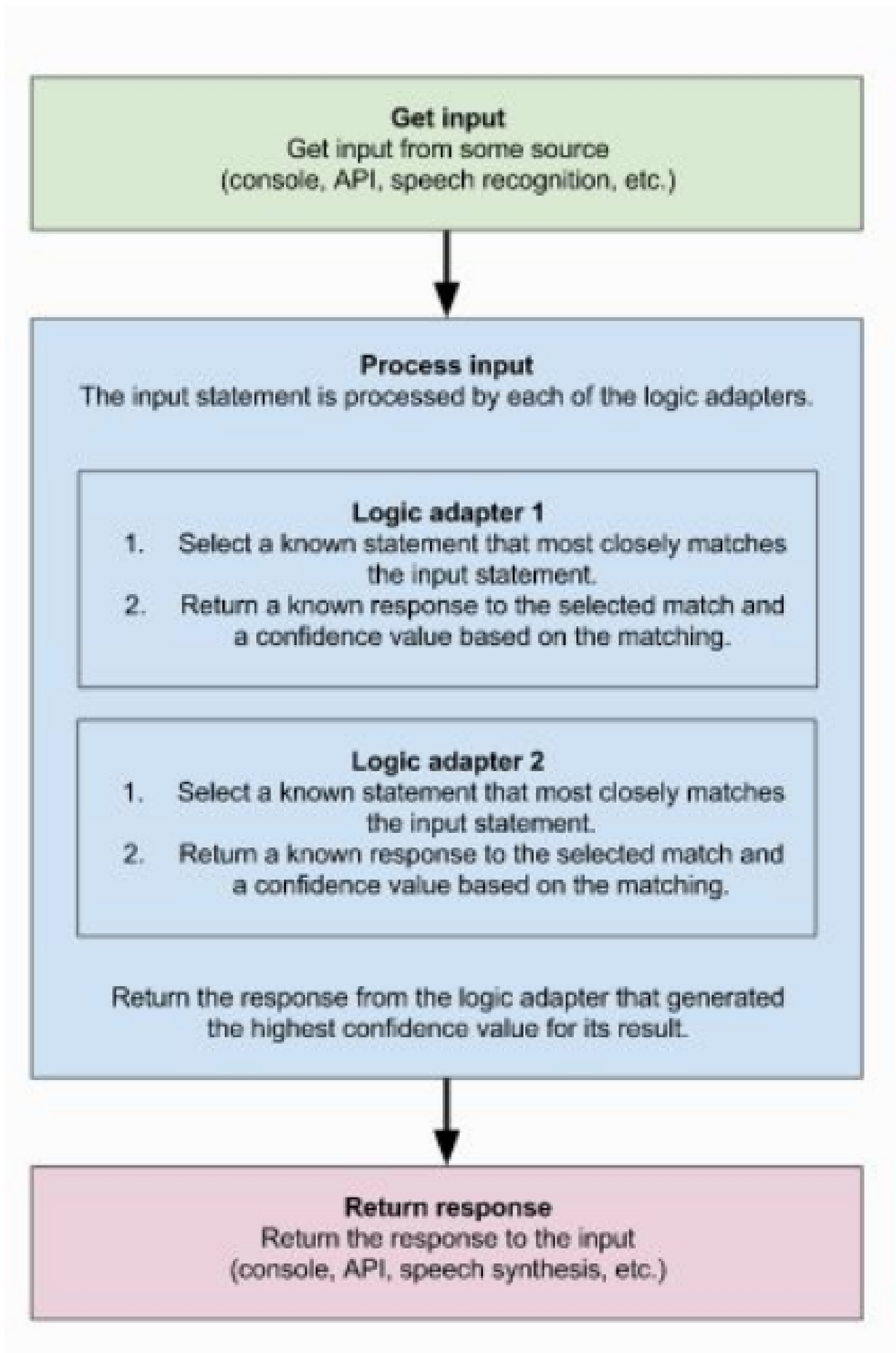

Fig: Process Flow Diagram of ChatterBot

8. Implementation

For the implementation of a 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 that simulates Golpo, a Bengali chatbot.
6. After the successful implementation of the system, we do a comparative study with the other two chatbots for testing.

Thus we have implemented the Bengali Chatbot.

9. Training

ChatterBot uses tools to make training a chatbot easier. It loads sample conversations into the bot's database, which creates or adds to the graph representing known statements and responses. When given a dataset, the trainer creates entries in the bot's knowledge graph to accurately represent input statements and their corresponding responses. ChatterBot includes built-in training classes for updating the bot's knowledge database and training it using conversation statements or preloaded data. Training Bengali Chatbot can occur in two steps.

- 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 strings.
- 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.

To utilize the Corpus Trainer Class in Chatterbot, the initial step involves creating a Bangla corpus in JSON format within the Chatterbot's data folder. Following this, the trainer is set to utilize the Bangla Corpus for training based on a structured format specified in the library. The provided pseudocode outlines the Corpus Trainer Class, originally developed by Gunther Cox, offering insight into its functionalities.

10. Storage Adapters

ChatterBot includes adapter classes for connecting to various databases. We'll use the Json File Storage Adapter, which stores data in a JSON file on the hard disk, ideal for testing and debugging. This adapter can be selected in the chatbot's constructor. The 'database' parameter defines the path to the database (database.json) that the chatbot will utilize, automatically creating the file if it doesn't exist.

11. Input Adapters

ChatterBot's input adapters are meant to handle various input sources. Parameters specifying input and output terminal adapters are necessary. The input terminal adapter reads user input from the terminal. The input adapter class, an abstract class, defines the interface for all adapters. Its main task is to classify input as known or unknown and label it accordingly before passing it to the logic adapter. The adapter's role is to convert input from different sources into ChatterBot's understandable format using the Statement object from the conversation module. In Golpo's implementation, a variable input adapter was used, enabling the chatbot to accept strings, dictionaries, and statements interchangeably.

12. Output Adapters

The output adapter enables the chatbot to generate a response as a Statement object. It's a flexible class that can be modified by a subclass to offer additional features, like sending a response to an API endpoint.

13. Logic Adapters

Certainly! The logic adapters within ChatterBot are responsible for determining how the bot selects responses to user input. The selection of a specific logic adapter for the bot can be done by setting the logic_adapters parameter to the desired logic adapter's import path. Multiple logic adapters can be utilized simultaneously, and if several adapters are used, the bot will return the response with the highest confidence value calculated by these adapters. In case multiple adapters return the same confidence level, the priority is given to the adapter that appears first in the list.

Among the available logic adapters, the "Best Match Adapter" is employed for our chatbot. This particular logic adapter functions by determining a response based on the closest matches to the input statement from the known responses in the database. It identifies the best match to a given input statement and then utilizes a function to select an appropriate response from the known responses associated with that matched statement.

The Best Match Adapter utilizes the Jaccard Similarity function to compare the input statement with the known statements. Jaccard Similarity compares two sentences using the Jaccard Index, which is a measure of similarity between two sets. It's computed as a ratio involving the numerator and denominator, essentially a fraction. In the numerator, the count represents the number of items shared between the sets, while the denominator indicates the total count of items across both sets.

For instance, consider two sample sentences: "The young cat is hungry." and "The cat is very hungry." After removing stop words, these sentences result in the following sets: {young, cat, hungry} and {cat, very, hungry}. The intersection of these sets, i.e., the common elements, is {cat, hungry}, with a count of two. The union of the sets, representing all unique elements, is {young, cat, very, hungry}, with a count of four. Therefore, the Jaccard similarity index in this example is calculated as two divided by four, resulting in 50%.

As per the threshold specified earlier (considering sentences equivalent if 50% or more of their tokens are equivalent), this comparison would be considered a match.

In summary, the Best Match Adapter within ChatterBot employs the Jaccard Similarity function to compare input statements with known responses, selecting the best match based on a specified similarity threshold.

14. Statement-Response Relationship

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

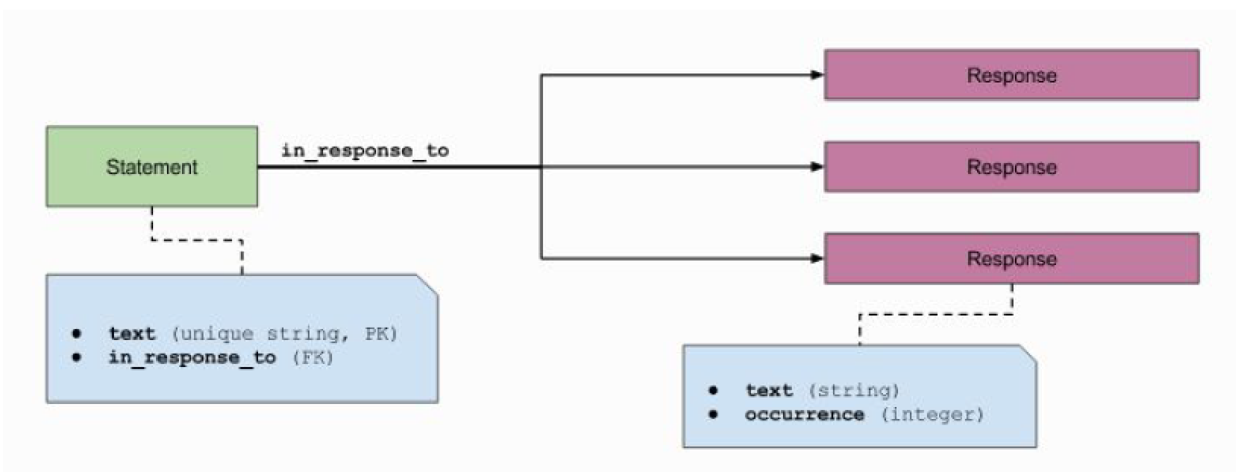

Fig: The Relationship between Statement and Responses

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.

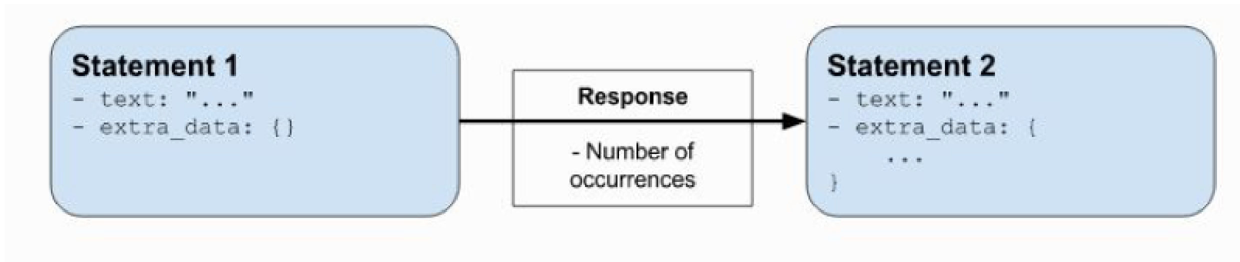

Fig: Mechanism of the reference to all parent statements of the current Statement.

The occurrence attribute of the Response object shows how many times the statement has been provided as a response. This enables the chatbot to ascertain whether a specific response is more frequently utilized than another.

15. 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 "
        ],
    ]
}
```


Fig: Format for Creating the Corpus for JSON Storage

16. Comparison between English Chatbot and Bengali Chatbot

Aspect	English Chatbot	Bengali Chatbot
Availability	Widely available and supported by various platforms and developers.	Less common and still under development. Limited availability across platforms.
Training Data	Vast amounts of training data available in English, leading to more accurate and sophisticated responses.	Limited training data available in Bengali, resulting in potentially less accurate and nuanced responses.
Technical Complexity	Building an English chatbot is generally easier due to the abundance of resources and tools available.	Building a Bengali chatbot requires additional effort due to the lack of resources and technical challenges in Bengali language processing.
Accessibility	Accessible to a wider audience globally due to the prevalence of English.	Primarily accessible to Bengali speakers, potentially limiting its reach.
Cultural Relevance	May not always be culturally relevant to Bengali speakers and may struggle with nuances of the language.	Offers a more culturally relevant experience by understanding and responding to Bengali language and cultural references.
Economic Potential	Offers wider economic opportunities due to the larger English-speaking market.	Has the potential to contribute to the development of the Bengali language and economy.
Educational Value	Can be used to promote English language learning and education.	Can be used to promote Bengali language learning and education, especially for non-native speakers.

Fig: Comparison between Bangla and English Chatbot

17. Analysis

We created and annotated a conversational corpus in Bengali manually because we couldn't find a suitable database for our purpose. We aimed to develop a Bengali chatbot despite the lack of necessary language processing tools like Parts of Speech Taggers and Tokenizers. To overcome this, we opted for a language-independent platform and chose a retrieval-based model. In evaluating our work, the absence of a benchmark for Bengali conversational agents posed a challenge. Therefore, we compared our Bangla chatbot with two popular English chatbots, Neural Conversational Model (NCM) and Cleverbot. To ensure a fair comparison, we translated the questions asked in Bengali to English and compared the responses.

NCM is an open-domain generative chatbot based on neural networks, while our Bangla chatbot operates on a closed-domain retrieval model. Interestingly, our system performed better than Cleverbot in many instances. We tested the Bangla chatbot with unknown sentences and found that initially, it produced random responses. However, it stored and reused user-provided responses, indicating functionality. It responds promptly in Bengali, showcasing effective pattern-matching algorithms. Our chatbot generates grammatically correct Bengali with accurate spelling, albeit with occasional punctuation mistakes that could be improved.

From the samples, it's evident that our Bangla chatbot produces similar responses to NCM and outperforms Cleverbot in most cases. Despite its strengths, the system lacks a coherent personality, which makes passing the Turing test challenging.

Our work's primary contribution lies in generating a conversation corpus in Bengali, which holds numerous advantages. A corpus is fundamental for language analysis and research across various foreign languages. This marks a significant shift ideologically and technologically in language research, facilitated by the integration of computers and corpora in linguistic studies. The Bengali language corpus can serve as a vital resource for technological advancements and linguistic studies, enabling the development of sophisticated automatic tools and systems while enriching language descriptions and theoretical frameworks.

18. Conclusion

In Bengali dialogue systems, this research is a significant study. Developing a chatbot based on a precise knowledge base is the primary task of this project. A retrieval-based closed-domain chatbot that uses a pattern-matching algorithm to communicate with users and learns from those interactions to improve its performance measure was built because a large dataset was unavailable. To aid in the creation of resources for Bengali Language Processing research, our study will produce a corpus of conversations in Bengali.

19. Future Work

The future work in this area could focus on several key aspects. Firstly, enhancing the Bangla chatbot's responses to unknown sentences by implementing more sophisticated algorithms for better contextual understanding and meaningful replies. Improving its punctuation accuracy and integrating a more coherent personality could enhance its performance in passing the Turing test. Additionally, developing a more extensive and diverse conversation corpus in Bengali would be crucial for refining the chatbot's language comprehension and response generation. Furthermore, exploring methods to expand its capabilities beyond a closed-domain retrieval model, possibly integrating elements of open-domain conversational AI, could broaden its conversational scope and depth. Lastly, establishing benchmarks and evaluation metrics specifically tailored for Bengali conversational agents would be essential for accurately assessing and comparing their performance.

20. 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 the WhatsApp 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)