

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/347902940>

Chatbot for Education System

Article · December 2020

CITATIONS

104

READS

25,477

2 authors, including:



Dr. K S Wagh

Savitribai Phule Pune University

46 PUBLICATIONS 284 CITATIONS

SEE PROFILE



INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X

Impact factor: 4.295

(Volume 4, Issue 3)

Available online at: www.ijariit.com

Chatbot for education system

Guruswami Hiremath

guru.h121@gmail.com

AISSMS Institute of Information Technology,
Pune, Maharashtra

Aishwarya Hajare

aishwaryahajare12@gmail.com

AISSMS Institute of Information Technology,
Pune, Maharashtra

Priyanka Bhosale

priyanakaarbhosale@gmail.com

AISSMS Institute of Information Technology,
Pune, Maharashtra

Rasika Nanaware

nanawarerasika8@gmail.com

AISSMS Institute of Information Technology,
Pune, Maharashtra

Dr. K. S. Wagh

kishorwagh@gmail.com

AISSMS Institute of Information Technology,
Pune, Maharashtra

ABSTRACT

The purpose of this paper is to develop an automated system which gives a reply to a user query on behalf of a human for the education system. It can give an answer to each and every query asked by the end user. Existing chatbots such as Facebook chat bot, WeChat, Natasha from Hike, Operator, etc. were giving reply from its local database. But our approach is to focus on the local database as well as web database and also to make system scalable, user-friendly, and highly interactive. Various techniques such as machine learning, NLP, pattern matching, data processing algorithms are used in this paper to enhance the performance of the system.

Keywords: Pattern Matching, NLP, Machine learning, Response generation.

1. INTRODUCTION

The development of the information technology and communication has made artificial intelligent systems more complex. The AI systems are approaching human activities such as taking a decision at a particular moment, performing day to day tasks. In an artificial intelligent field, there are some hybrid methods and adaptive methods available which are making systems more complex. Not only that but also there is a hybrid combination of natural language processing and intelligent systems. These systems can learn themselves and renew their knowledge by reading all electronics articles available on the internet. Human as a user can ask the systems like usually ask another human. These systems are often known as internet answering engines. In addition to the internet answering engines, currently, many applications are introduced such as chatter-robot or known as chatbot which is often aimed at giving an automatic reply or just for entertainment. This application's work is very simple because the knowledge is already programmed in advance. Few of the methods used in this application are pattern-matching, natural language processing, data mining. The chatbot would match the input sentence from the speaker or user with that pattern existed in the knowledge base. Each pattern is then compared with the knowledge of chatbot. This knowledge has been taken from various sources.

2. METHODS AND MATERIALS

2.1: Literature Survey: By using Artificial Intelligence field, we can develop many applications one of that is mentioned in this paper is chat bot for the education system. Though chat bot has a variety of fields like marketing, medical, education, banking, and finance. We will see related work in each field. Medical field chat, but is useful for providing free counseling to the patient, suggesting various medicines, etc. Even without going to the hospital or visiting the doctor. Kyo-Joong oh & Dong Kun Lee in [1] have proposed a chat bot psychiatric counseling in mental health care service. Emotion recognition is used to implement this chat

bot. Because of this, the patient is not needed at all to go to the hospital. Free counseling is provided at the user's location. Nikita Hatwar in [2] proposed another interesting chat, but for marketing field named as AI based chat bot. In this paper, this chat bot is used for providing guidance to the visitor of the mall, provides navigation and according to latest discount going in the shop, it was guiding recommendation system is implemented in this project according to the rating we can advise people to go to a particular shop. This feature can be extended further. In the marketing field, the user has lots of questions related to projects whose answer is not present even in the local database, to overcome this problem Prof. Yu Wo and Gongxiao Wang in [3] presented paper automatic chat bot knowledge acquisition from online via rough set and ensemble learning. The concept of ensemble learning is used here. The constructs classification results of the learner to get the final result. Multiple rough set, classifier is constructed and trained first, and then all replies are classified with these classifieds. The final results are drawn from voting to the output of these classifiers and finally, it is selected as a knowledge database. The disadvantage of this system is that not all replies are related to root message different forum have different styles and formats. The education system also has chat, but Naveen Kumar and Linga chandar P.C in [4] have proposed an android based educational chat bot for visually impaired people. Voice processing, symbolic reduction, and keyword detection methodologies are used.

In case of noisy environment, the application is not recommended. The application should be scalable, so even if thousands of queries are asked at a time. Then also system is able to answer all the queries. Juan Carlos, et al [5] proposed a system that is chatting with an Arduino platform through the telegram boy. The proposed system is connecting telegram and Arduino platform using telegram boy. This system allows people to communicate with machine user, where this machine user can operate simple and complex prototype developed by the same person. To implement this system, telegram apes are used. REST protocol is used for the integration of both. For Social networking field Jacques Gracia Fausto Abraham have proposed and developed a chatbot in a social network in [6], This paper describes how to implement chat bot on twitter for entertainment and viral advertising using the database and simple algorithms. Message reception, message passing, message processing techniques are used. With this system, a person can earn more followers every day without any human intervention. In social networking security is the main concern to implement this functionality, Wenzeng Zhu in [7] proposed a system that is internet security protection for IRC based BotNet many works contain infected data which can harm our system. So in this paper, they proposed a noble paper based on IRC analysis and detection to protect our computers. IRC servers, used instead of C&C server. So the infected computers link to the IRC server. This method can protect the infected computers not to return Bots controlled by the Bot Master. The only problem with this is that we need to detect all the infected computers and help them to uninstall bot code. To return securely, but the problem is that some boot code is hard to uninstall. From the above survey, it is to be concluded that none of the authors is focusing on making chat bot application secure and scalable. By encrypting user query we can secure chat bot and using the Apache Mahout we can make chat bot scalable.

2.2 Algorithms:

2.1. Input Adapters:

Chatbot's input adapters are designed to allow a chat bot to have a versatile method of receiving or retrieving input from a given source. Creating a new input adapter:

The goal of an input adapter is to get input from some source, and then to convert it into a format that Chatbot can understand. This format is the Statement object found in Chatbot's conversation module.

This adapter accepts strings, dictionaries, and Statements.

```
chatbot = ChatBot(  
    "My Chatbot",  
    input_adapter="chatbot.input.VariableInputTypeAdapter"  
)
```

Terminal input adapter

```
chatbot.input.terminal adapter(**kwargs)[source]
```

A simple adapter that allows Chatbot to communicate through the terminal. The input terminal adapter allows a user to type into their terminal to communicate with the chat bot.

```
chatbot = ChatBot(  
    "My Chatbot",  
    input_adapter="chatbot.input.TerminalAdapter"  
)
```

2.2 Logic Adapter: Logic adapters determine the logic for how Chatbot selects a response to a given input statement.

1) The MultiLogicAdapter

2) How logic adapters select a response

2.2.1 The MultiLogicAdapter :

Chatbot internally uses a special logic adapter that allows it to choose the best response generated by any number of other logic adapters. Selecting a response from multiple logic adapters. The MultiLogicAdapter is used to select a single response from the responses returned by all of the logic adapters that the chat bot has been configured to use. Each response returned by the logic adapters includes a confidence score that indicates the likeliness that the returned statement is a valid response to the input.

2.2.1.1 Response selection:

The MultiLogicAdapter will return the response statement that has the greatest confidence score. The only exception to this is a case where multiple logic adapters return the same statement and therefore agree on that response. For this example, consider a scenario where multiple logic adapters are being used. Assume the following results were returned by a chat bot's logic adapters.

Confidence	Statement
0.2	Good morning
0.5	Good morning
0.7	Good night

In this case, two of the logic adapters have generated the same result. When multiple logic adapters come to the same conclusion, that statement is given priority over another response with a possibly higher confidence score. The fact that the multiple adapters agreed on a response is a significant indicator that a particular statement has a greater probability of being a more accurate response to the input. When multiple adapters agree on a response, the greatest confidence score that was generated for that response will be returned to it.

```
class chatbot.logic.multi_adapter.MultiLogicAdapter(**kwargs)[source]
```

MultiLogicAdapter allows Chatbot to use multiple logic adapters. It has methods that allow Chatbot to add an adapter, set the chat bot, and process an input statement to get a response.

add_adapter(adapter, **kwargs)[source]

Parameters: adapter (LogicAdapter) – The logic adapter to be added.

get_adapters()[source]

Return a list of all logic adapters being used, including system logic adapters.

get_greatest_confidence(statement, options)[source]

Returns the greatest confidence value for a statement that occurs multiple times in the set of options.

Parameters:

statement – A statement object.

options – A tuple in the format of (confidence, statement).

get_initialization_functions()[source]

Get the initialization functions for each logic adapter.

insert_logic_adapter(logic_adapter, insert_index, **kwargs)[source]

Adds a logic adapter at a specified index.

Parameters:

logic_adapter (str) – The string path to the logic adapter to add.

insert_index (int) – The index to insert the logic adapter into the list at.

process(statement)[source]

Returns the output of a selection of logic adapters for a given input statement.

Parameters:

statement – The input statement to be processed.

remove_logic_adapter(adapter_name)[source]

Removes a logic adapter from the chat bot.

Parameters:

adapter_name (str) – The class name of the adapter to remove.

set_chatbot(chatbot)[source]

Set the chatbot for each of the contained logic adapters.

2.2.1.2 How logic adapters select a response:

A typical logic adapter designed to return a response to an input statement will use two main steps to do this. The first step involves searching the database for a known statement that matches or closely matches the input statement. Once a match is selected, the second step involves selecting a known response to the selected match. Frequently, there will be a number of existing statements that are responses to the known match. To help with the selection of the response, several methods are built into ChaterBot for selecting a response from the available options.

Response selection methods

chatbot.response_selection.get_first_response(input_statement, response_list)[source]

Parameters:

input_statement (Statement) – A statement, that closely matches an input to the chat bot.

response_list (list) – A list of statement options to choose a response from.

Return the first statement in the response list.

chatbot.response_selection.get_most_frequent_response(input_statement, response_list)[source]

Parameters:

input_statement (Statement) – A statement, that closely matches an input to the chat bot.

response_list (list) – A list of statement options to choose a response from.

returns the response statement with the greatest number of occurrences.

chatbot.response_selection.get_random_response(input_statement, response_list)[source]

Parameters:

input_statement (Statement) – A statement, that closely matches an input to the chat bot.

response_list (list) – A list of statement options to choose a response from.

Returns: Choose a random response from the selection.

2.3. Output Adapters: It is used to generate an output which will be displayed to the user. To create the output adapter use the following function:

chatbot.output.OutputAdapter(kwargs)[source]**

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

```
chatbot = ChatBot(  
    "My Chatbot",  
    output_adapter="chatbot.output.OutputAdapter",  
    output_format="text"  
)
```

2.4 Training System:

Chatbot includes tools that help simplify the process of training a chat bot instance. Chatbot'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.

2.4.1 Chatbot training statement graph

Several training classes come built-in with Chatbot. 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 pre-loaded training data. You can also create your own training class. This is recommended if you wish to train your bot with data you have stored in a format that is not already supported by one of the pre-built classes listed below.

2.4.2 Setting the training class

Chatbot comes with training classes built in, or you can create your own if needed. To use a training class you must import it and pass it to the set_trainer() method before calling train().

Training classes

Training via list data

chatbot.trainers.ListTrainer(storage, **kwargs)[source]

Allows a chat bot to be trained using a list of strings where the list represents a conversation.

For the training process, you will need to pass in a list of statements where the order of each statement is based on its placement in a given conversation.

For example, if you were to run bot of the following training calls, then the resulting chatbot would respond to both statements of “Hi there!” and “Greetings!” by saying “Hello”.

```
from chatbot.trainers import ListTrainer
```

```
chatbot = ChatBot("Training Example")
```

```
chatbot.set_trainer(ListTrainer)
```

```
chatbot.train([
```

```
    "Hi there!",
```

```
    "Hello",
```

```
])
```

```
chatbot.train([
```

```
    "Greetings!",
```

```
    "Hello",
```

```
])
```

You can also provide longer lists of training conversations. This will establish each item in the list as a possible response to its predecessor in the list.

```
chatbot.train([
```

```
    "How are you?",
```

```
    "I am good.",
```

```
    "That is good to hear.",
```

```
    "Thank you",
```

```
    "You are welcome.",
```

```
])
```

3. WORKING

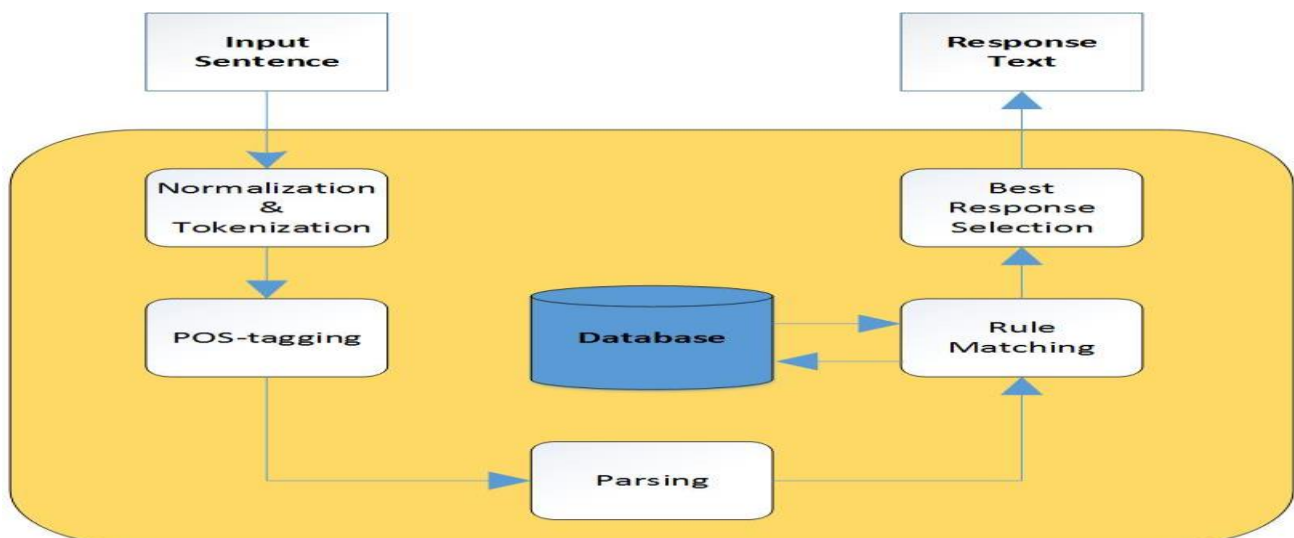


Fig 1. Working of Chatter Bot

ChaterBot includes tools that help simplify the process of training a chat bot instance. ChaterBot'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.

4. RESULTS

The system's output is shown in following figures.

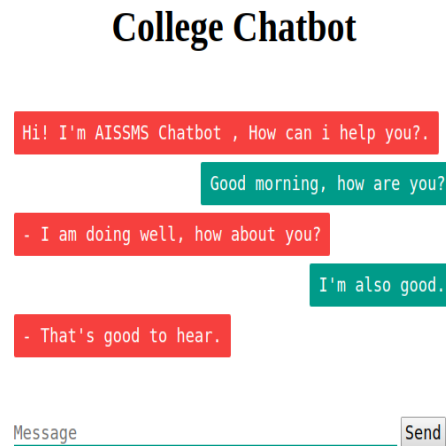


Fig 2: Greeting chatbot

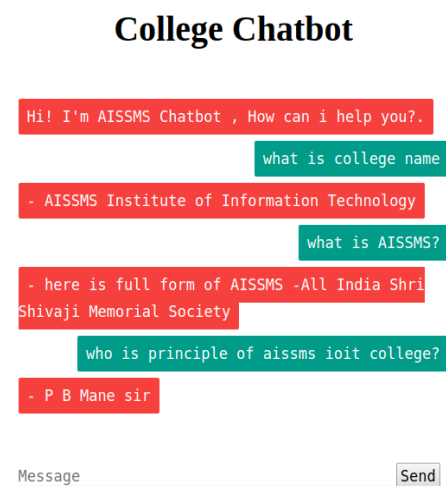


Fig 3: Chatbot College details

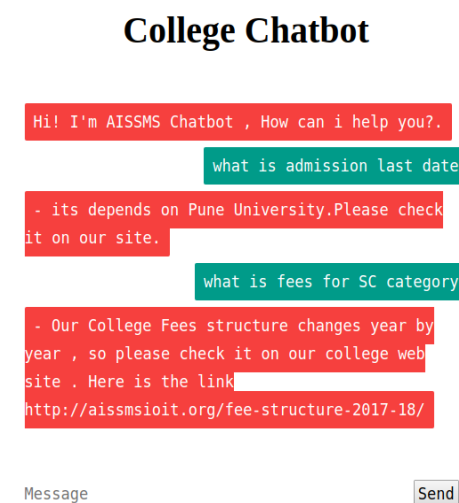


Fig 4: Chatbot Admission or Fees related query

5. CONCLUSION

In this paper, we have implemented an automatic response giving system which will give a reply to the student's questions. The use of artificial intelligence and machine learning is used for implementing this system. The user will type his or her queries and then the system will extract the proper keyword from the given query and will produce the response. The future scope can be implemented in this project in the following manner: If data is not available in a static database then it will be fetched from online sources. Because of this, every answer of the user will be generated either from online sources or static database. So in this way we have implemented an automated response generation system.

6. REFERENCES

- [1] K. J. Oh, D. Lee, B. Ko and H. J. Choi, "A Chatbot for Psychiatric Counseling in Mental Health Care Service Based on Emotional Dialogue Analysis and Sentence Generation," 2017 18th IEEE International Conference on Mobile Data Management (MDM), Daejeon, 2017, pp. 371-375
- [2] Prof. Nikita Hatwar "chat bot for marketing field named as AI based chat bot." In 2016
- [3] Y. Wu, G. Wang, W. Li, and Z. Li, "Automatic Chatbot Knowledge Acquisition from Online Forum via Rough Set and Ensemble Learning," 2008 IFIP International Conference on Network and Parallel Computing, Shanghai, 2008, pp. 242-246.
- [4] M. N. Kumar, P. C. L. Chandar, A. V. Prasad and K. Sumangali, "Android-based educational Chatbot for visually impaired people," 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Chennai, 2016, pp. 1-4.
- [5] J. C. De Oliveira, D. H. Santos, and M. P. Neto, "Chatting with Arduino platform through Telegram Bot," 2016 IEEE International Symposium on Consumer Electronics (ISCE), Sao Paulo, 2016, pp. 131-132
- [6] Randall Wald, Taghi M. Khoshgoftaar, Amri Napolitano, Chris Sumner, "Predicting susceptibility to social bots on Twitter", Information Reuse and Integration (IRI) 2013 IEEE 14th International Conference on, pp. 6-13, 2013.
- [7] W. Zhu and C. Lee, "Internet security protection for an IRC-based botnet," 2015 IEEE 5th International Conference on Electronic Information and Emergency Communication, Beijing, 2015, pp. 63-66.