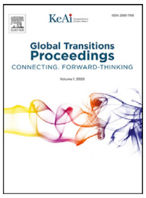
Contents lists available at ScienceDirect

Global Transitions Proceedings

journal homepage: http://www.keaipublishing.com/en/journals/global-transitions-proceedings/



Agroxpert - Farmer assistant

Vandana Nayak[a,*], Pranav R Nayak N[b], Sampoorna[c], Aishwarya[d], N.H. Sowmya[e]

[a] Dept. of Computer Science and Engineering, Shri Madhwa Vadiraja Institute of Technology and Management (Affiliated to VTU), Udupi, India-574115.
[b] Dept. of Computer Science and Engineering, Shri Madhwa Vadiraja Institute of Technology and Management (Affiliated to VTU), Udupi, India-574115.
[c] Dept. of Computer Science and Engineering, Shri Madhwa Vadiraja Institute of Technology and Management (Affiliated to VTU), Udupi, India-574115.
[d] Dept. of Computer Science and Engineering, Shri Madhwa Vadiraja Institute of Technology and Management (Affiliated to VTU), Udupi, India-574115.
[e] Dept. Computer Science and Engineering, Shri Madhwa Vadiraja Institute of Technology and Management (Affiliated To VTU) Udupi, India-574115.

ARTICLE INFO

Keywords:
Agriculture
Chatbot
Chatterbot
Django
Farming assistant
Natural language processing(NLP)

ABSTRACT

Agriculture occupies an important position in the Indian economy. Indian farmers today are facing the problem of low income due to the lack of information about government schemes, fertilizers, farming equipment etc. Some smallholders and marginalized farmers have low awareness as most of them live in remote areas and don't have access to information about soil properties, seeds, recently used tools, fertilizers, etc. The document proposes an intelligent, portable system that uses natural language processing methods to help farmers use different farming methods, and further help them to answer their queries and solve their basic and intermediate level doubts using chatbot which will save their time. To meet all the requirements of farmers, a chatbot is proposed using natural language processing technology. The system will act as an interactive virtual assistant for farmers, answering all queries related to agriculture. This paper will go through the implementation of the chatbot using the chatterbot libraries and Django framework.

1. Introduction

Farming plays a major role in the development of the country. It provides employment opportunities to a large number of people around the world. Day-by-Day the technology is developing in the agriculture field [1]. Nowadays the government is collecting data regarding rainfall and crop production but it is of no use for farmers. Analysing and collecting the data can only be useful to farmers to provide relational trends. There is also some software to teach farmers about technologies. But most of these technologies are not feasible as they do not give accurate answers to the queries asked by the farmers [2].

There are farmers who live in remote areas of the country, who do not have any information related to new technological trends and agricultural practices, who do not have access to sufficient information on crops, soil properties, seeds, latest tools used, fertilizers, etc. Lack of access to agriculture knowledge and information about the latest farming practices leads to poor crop and livestock productivity. To overcome the drawbacks, a chatbot is proposed using the pattern matching technique that provides an interface, where the users, in this case farmers, can communicate with the chatbot efficiently. The project is a web-based application that has Farmer and Admin login for privacy purposes. This system "Agroxpert" is basically a chatbot, which acts as a farming assistant that clears all the doubts of the farmers in an efficient manner. Using Machine learning algorithms query responses are generated [1,2,5]. If the proposed system does not respond to the farmers' queries, these queries are forwarded to experts. Apart from farmers, it will also help the people who are beginners and interested in farming [1].In future sentiment analysis and language translation can be applied to this application [3,7]. The Chatbot has some unique features like adding the question and retraining, updating the existing question, mailing the unanswered queries to escalate those query to higher level via expert. These operation makes the chatbot reply efficiently.

2. Related work

Arora et al. [4] has proposed a chatbot which in addition to assisting the farmers, also aims at detecting crop diseases and weather prediction. For detecting crop diseases, a CNN model is used which would segregate the images of plants into various classes. Results are generated in telegram messenger naming the crop disease that infected it. Good results with an accuracy of 98% for the chatbot module and an accuracy of 94% for the crop detection module were obtained. The future scope would be to enable language translation feature for chatbot module and location feature for weather prediction module. In this the chatbot is created in the telegram app to find out the disease.

[*] Corresponding author: Dr. Vandana Nayak, Shri Madhwa Vadiraja Institute of Technology and Management (Affiliated to VTU), Dept. of Computer Science and Engineering, Udupi, 574115, India

E-mail addresses: nvandana1@gmail.com (V. Nayak), npranavr@gmail.com (P. R Nayak N), sampz2199@gmail.com (Sampoorna), aishwaryapoojary99@gmail.com (Aishwarya), sowmyanh.cs@sode-edu.in (N.H. Sowmya).

https://doi.org/10.1016/j.gltp.2021.08.016
Received 13 June 2021; Accepted 5 July 2021
Available online 12 August 2021

Kasthuri and Balaji [6] due to the pandemic situation, the mode of learning is shifted to online. A major disadvantage of online based learning is the lack of student-teacher interaction. Hence, the main objective of this paper is to build a chatbot where students can ask any question and the chatbot responds with exact answers using NLP and deep learning techniques.

Khin and Soe [8] chatbot, a conversational agent, is widely used today in banking, marketing, education etc. So, here a University chatbot is built that gives efficient responses to any queries asked by users regarding University information. This chatbot is implemented using Artificial Intelligence techniques. It uses Myanmar language to communicate the information between the University and the users.

Koundinya K. et al. [9] has built an interactive chatbot for the users, mainly students, to access college websites using NLP techniques. This chatbot is designed in such a way that it incorporates human appearance and the students can communicate with the bot in an effective way and the chatbot responds with an accurate answer as it is trained well using some of the ML algorithms.

Vamsi et al. [10] proposes a new method of building chatbot using AI and deep learning. The chatbot is built in such a way that it should be able to understand users' perspective and then deliver accurate answers. Here, multiple layered neural networks are constructed in order to process the data.

Bhartiya et al. [11] proposes a University Counselling chatbot which responds to the queries related to the University. It makes use of NLP techniques and the problem of overfitting was handled effectively. This bot is then deployed on Facebook Messenger providing a useful, interactive platform for the users.

Kiruthika et al. [12] provides an interactive chatbot for direct marketing between the farmers and the consumers at a reasonable price as it does not require the involvement of a middleman. Here, farmers can communicate with consumers directly as they can access the chatbot via any social media such as facebook. With the advent of this chatbot, the consumer can now hire the farmer and provide all the necessary inputs; once the product is ready the farmer delivers the fresh product to the consumer thereby getting a good pay.

Gunawan et al. [13] proposes a system where the users want to know about the plant condition and the chatbot responds to the users' query by applying NLP techniques on the sensor data. Here, IOT is used to inform the users regarding plant conditions so that they can take necessary steps such as watering the plants etc.

Hristidis [16] discusses the technologies used in the chatbot, differences between transactional and conversational chatbots, and tools and software used for building chatbots. It also tells briefly about the disadvantages and future work needed for creating an effective chatbot.

Xie et al. [19] proposed an interactive chatbot on cryptocurrency using the Chatterbot library. In the current digital era, many investors are very enthusiastic to know everything about the new digital currency. Based on the queries asked by the user, Itchat API will respond to it and its response time is compared. Future enhancement would be to improve the implementation of the logic adapter.

3. Methods

There are various ways to create a customised chatbot. One such method is Chatterbot. Chatterbot is the python library used to create the chatbot application and it is based on machine learning. It generates a response based on the collection of responses or conversation.

3.1. Data collection

For Working of any chatbot the data is important for training the chatbot. Therefore the main step of developing the chatbot is data. The knowledge base for the chatbot is obtained from the data that is collected. Grouped data from the most asked question and answers about farming or agriculture from the internet. This gathered data is given as a dataset for training the model. The dataset contains two attributes: query (question) and response (answer) in CSV format as shown in Table 1.

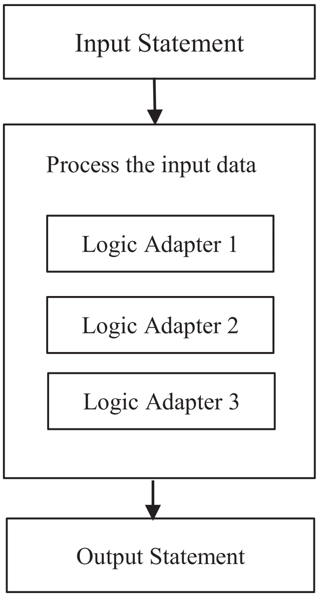


Fig. 1. Working of the chatterbot library and processing of the input.

In future the expert will add the possible question and answer depending on the type of query from the farmer because there are many possible questions and it can be changed from time to time so as there is a need for an update of the chatbot and it can handle all types of queries from the farmer. The query present in the dataset represented in comma format indicates different possible ways of asking questions having same response (answer).

3.2. Tools used

The chatbot is developed using python in the backend which handles all the server related activities using the Django framework of the python. Django is installed using the command "pip install Django".Front end is built using HTML, CSS, JavaScript, jQuery and AJAX.

3.2.1. Chatterbot library

The tool used to build the chatbot is chatterbot. Chatterbot is trained with the known responses for the questions. After training the chatterbot, the chatbot stores the questions and answers with the best possible match. After the training, the Chatterbot is ready to answer the questions to which chatterbots have already been trained.

Input for the chatterbot is in the form of text. The farmer asks the question to the chatbot through a web interface and the asked question is processed through logical adapters. In chatterbot, logic adapter is a class that takes an input statement and returns a response to that statement. You can have any number of logic adapters. Let's take the example of two logical adapters as given in Fig. 1. The given input is sent to the first logical adapter, say, time logic adapter; it checks if the question needs the time as a response to it and gives the response to the next logical adapter. Let's assume that the second adapter is a mathematical evaluation adapter. The adapter checks if the response requires mathematical calculation and the adapter calculates the expression and gives the required result as a response. There are so many other similar adapters. One of the important adapters is the maximum

Similarity threshold, this is to match the dataset question with user input questions. The response is sent only if the similarity of the dataset and the user input is more than the maximum similarity index.

Table 1
Example of dataset used in the chatbot application.

Question / Query	Answer / Response
Hello, Hi	Hi, I'm Agroxpert. You can ask me about agriculture
How many varieties of rice are grown in India?	Approximately 6,000 different varieties, but also lost tens thousands of varieties from last 40 years
Which is the most popular rice grown in India?	Basumati rice

3.2.2. Web interface

It is a web-based chatbot. Hence, by using HTML, CSS, JavaScript, jQuery, AJAX front end is created, and Django, Python, MySQL in the backend. As it is a web application it can be accessed on any device which has the browser. The Application is built using the Intel I3 processor, Ram 4GB, Notepad, and Sublime text.

3.2.3. Xampp with PHPMyAdmin

Xampp is used in the project for the database. For working with the PhpMyAdmin, apache server is required. Xampp provides both PHPMyAdmin and apache for local computers and it is open source and free to use. PhpMyAdmin is required to use the database for login and other purposes.

3.2.4. MySQL-connector

To connect python with the database, MySQL connector is required. To work with the latest version it requires MySQL server version 8.0, 5.7, 5.6, 5.5 and Python version 3.9. To install the MySQL connector, use command in the command prompt.

3.3. Design and architecture

Chatbot designing process begins by identifying the target. Here the target is a farmer, who can ask and clear his doubts with the chatbot. After understanding this the data collection was the next step. Gathering more data and training the chatterbot, more efficient the chatbot would become. Now as the data is collected, the chatbot will be built using chatterbot. Later implementation was done to a user interface for friendly use. One additional user has been included called Admin or expert, who can add more questions to the dataset using the interface.

The proposed chatbot uses Levenshtein distance formula for calculating the difference in the string and returns a value which is named as confidence value.

3.3.1. Levenshtein distance

Levenshtein distance is defined as the value that describes the minimum change required from input into output. It is used in natural language processing. Mathematically, the Levenshtein distance can be calculated as follows,

If min(i,j) ≠ 0,

$$lev_{a,b}(i,j) = min\{lev_{a,b}(i-1,j)+1\ lev_{a,b}(i,j-1)+1\ lev_{a,b}(i-1,j-1) + 1_{(if\, a_i \neq b_j)} \tag{1}$$

Others,

$$lev_{a,b}(i,j) = max(i,j) \tag{2}$$

In equation number (1) and (2), *a* and *b* represent the two strings where the Levenshtein distance needs to be found out where as i and j represent the terminal characters of the string a and b, respectively. The similarity value between two sentences is based on the Levenshtein distance. The greater the Levenshtein distance, the greater the difference between the strings. The value ranges between 0 and 1. For example, the Levenshtein distance from "chatbot" to "chatbot" is 0, because the source and target strings are the same, and the sentence similarity value is 1. The string "chatbot" to "chatbat", the distance is 1, because of the difference between the strings and its similarity index is 0.86. Here, the sentence similarity value is calculated based on the Levenshtein distance. The lower the score, the less similar the statements, as shown in Table 2.

Table 2
Sentence similarity value using Levenshtein distance.

Statement 1	Statement 2	Similarity Value
Chatbot	Chatbot	1
Chatbot	Chatbat	0.86
Where is the post office?	Check for the post office?	0.65
Dance	Swim	0

3.3.2. Best match

Best match algorithm is used to find the output of the question asked from the user. This algorithm takes input as a statement or question and provides a response as the output. *: Best match*

```
input:  statement
output: response statements ← statements with the known response;
  for statement2 ∈ statements do confidence score ← Levenshtein distance between statement and statement2;
  if confidence score is higher then statement with highest score ← statement2; responses ← get the responses with statement with highest score;
  if more than zero responses then get the first element from responses;
```

The input statement is taken from the user and the output is returned to the user. The statement and the response both will be stored in “statements” variables from the training dataset. Then comparing the statement given by the user and each statement from the data set using Levenshtein distance formula and the confidence level of that statement is calculated. If the statement’s confidence value is greater than the default confidence value then the statement is returned by the best match algorithm. The best match algorithm is used in the chatterbot library where it is used to find the response for the query asked.

3.3.3. Sample code

To install the chatterbot use pip command

“pip install chatterbot.”

To run the chatterbot create a new file named chatbot.py. Then open chatbot.py in any editor.

Following is the example code for building a terminal running chatbot:

```
from chatterbot import ChatBot
bot = ChatBot(’Agroxpert’)
from chatterbot.trainers import ListTrainer
trainer = ListTrainer(bot)
trainer.train([
’Hi, hello ’,
’Hello, this is Agroxpert. How can I help you?’,
’Which soil type is good for coffee cultivation?’,
’Well drained loamy soil',
'In which season paddy is cultivated',
Paddy is cultivated in the rainy season',
```

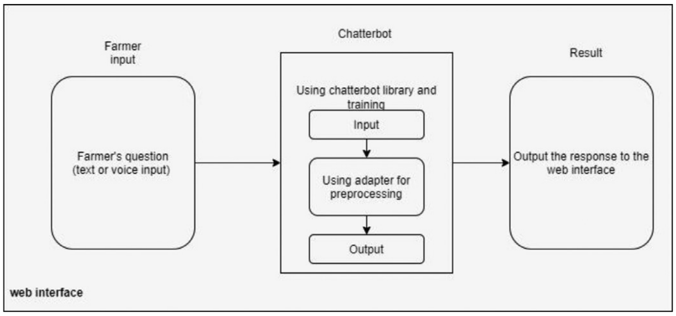

Farmer input
Farmer's question (text or voice input)
Chatterbot
Using chatterbot library and training
Input
Using adapter for preprocessing
Output
Result
Output the response to the web interface
web interface

Fig. 2. System architecture.

```
  ’Thank you’,
  'Welcome',
  ])
  while True:
  try:
  bot_input = bot.get_response(input())
  print(bot_input)
  except (KeyboardInterrupt, EOFError,
SystemExit):
  break;
```

Note: Adding the adapter and use of the database is optional.

The code will train the chatbot with four possible sets of query and response. Chatbot() is used to create new chatbots and the ListTrainer() will create an instance of the chatbot. get_response() method will fetch the answer which takes the question as an input to provide the suitable response.

3.3.4. Architecture

The User interface is implemented using HTML, CSS, JavaScript and backend on Django framework along with MySQL RDBMS for the storage of the queries and users login credentials. As part of the security, MD5 and base64 are used for encryption of password. The database is created in the MySQL named Agroxpert for handling the database related operation where the database mainly contains three tables which are Admin, Farmer and Mail. Admin table is used for storing the admin details whereas Farmer table is used to store the farmer details. Basic details of farmers and admin are taken like Name, ID, Password, phone number or email address. Mail table mainly consists of six fields namely, Mail ID, Farmer ID, Expert ID, Subject, Query and reply fields. Using the connect method from mysql-connector the connection is established with the database by having the basic details like server name, username, password and database. Once the connection is established using SQL insert query the data can be registered and using the SQL select statement the registered data can be fetched from the database.

There are two types of users in the Agroxpert those are farmers or expert. If the user is registered, then he can directly login else he needs to register himself as either farmer or expert as shown in Fig. 2.

Farmers chat with the chatbot to clear their doubts. Main purpose for building the chatbot is to provide the information to the farmers and solve their doubts quickly and at any time. Once the farmer sends any query from the User interface. Farmers can send the query either by typing or by using the voice to text conversion which is built using JavaScript's speech recognition function. Once the query of the farmer is fetched from the frontend using Ajax then the query is given as an input to the chatterbot using get_response method. Where it provides the response depending upon the available data. If the response is not available then default response is provided redirecting the farmer to admin.

Fig. 2 represents the working of the Agroxpert chatterbot for the chatterbot the input is provided from the front end just like in Fig. 6. For the farmer to ask the question he needs to be logged in just like in Fig. 5.After the login the farmer is eligible to ask the question with Agroxpert. The input box is provided to ask the question where the farmer gives the input in the form of text or voice. If the input is given in the form of the voice then it is automatically converted to text. After getting the input it is processed using chatterbot libraries and using different types of adapters like mentioned in Fig. 1. After processing the input in different adapters the result is given to be displayed using the browser.

Chatbot can sometimes fail to answer the farmer's question [2].When any farmer's doubts are unresolved then he can mail the expert for clearing the doubts, where experts can help them by suggesting the possible solutions to the farming or any related query. When any farmer uses mail functionality then from the farmer the subject or title and main issue is asked to the farmer and it is stored in the database using SQL insert query for table named mail. The same is forwarded to all the experts whoever is registered. Farmer ID, Subject and query field will be inserted and a unique chat ID is automatically generated for the same. Expert ID, reply field will be empty in the mail table in the database.

An expert or admin module is created. It has different functionalities. One functionality of an expert or admin is replying to a farmer's query. Upon the login of the admin it checks for the mail table in the Agroxpert database if any reply field is empty using SQL select query and where option. If the reply field is empty, then the same is displayed to the expert for their solution to the farmer's query. When an admin/expert replies to the farmer's query using the SQL update query the solution and Expert ID is inserted to the same field using the Chat ID which will notify the reply of the expert to a farmer. Second one is to add new questions and answers for training the chatbot for solving the new queries. When experts want to add any new query and response then both the query field and response field is mandatory whereas the URL field is optional. URL feature is added since whenever the query containing a link is asked and whenever the farmer clicks the link it redirects the user to a site where he may get the required detailed answer. If an expert adds any URL then it is converted to HTML anchor tag appending the URL provided by the expert in the href field. Using all this the chatterbot is trained using the train method of the ListTrainer of chatterbot. After training the same is reflected to the dataset. Third is to modify the existing queries. In case the expert makes some mistakes or there is change in any response to the query then they can use the update functionality. In this functionality he can edit the question, answer or URL once they edit then the data is trained in the backend. One drawback is that once edit is made the conversation which is stored by the chatterbot's storage adapter should be deleted using the drop method from the chatterbot.

4. Result and discussion

The web based application is built using Django to implement this model. Using MD5 the security on login and registration credentials has been achieved. The main purpose of building a web application over android application or GUI is that web application is easily accessible in any device which has a browser. Also, web application won't take much space in the user's device and can be easily maintainable. One limitation is stable internet connection is needed for the working of the web application.

Fig. 3 is the initial page whenever the user visits the website. The home page has an adorable animation and greets the user. It has a login dropdown which has two modules namely famer and admin.

Fig. 4. is the farmer registration page where farmers can register by giving all the necessary information. If he already has an account, he can switch to the login page and provide login credentials. If a farmer has already registered and tries registering again then there will be an error saying error in sign up.

Fig. 5 is the Farmer Sign in page where farmers have to login using farmer ID and password and in case if he does not have an account, he needs to create using farmer's registration form. The admin also has a

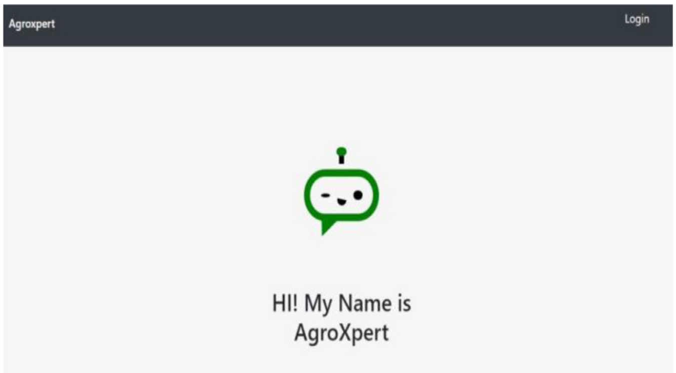


Fig. 3. Home page.

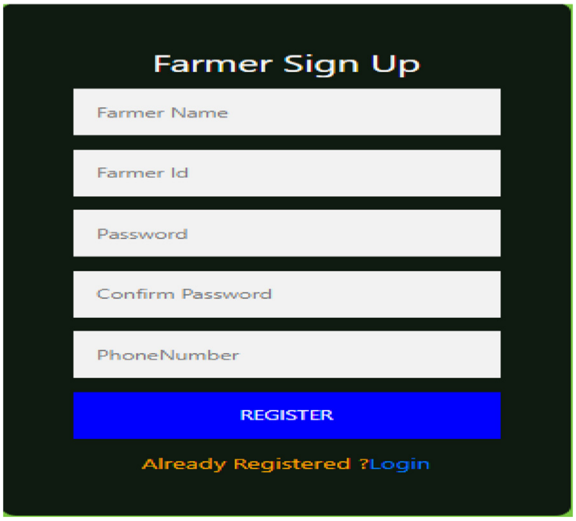


Fig. 4. Farmer registration page.

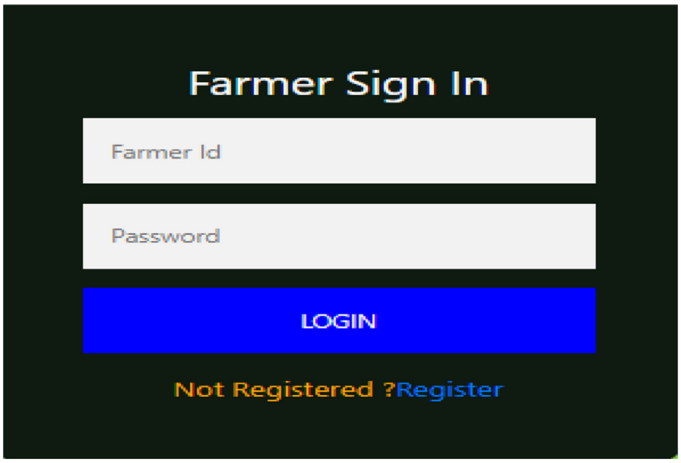
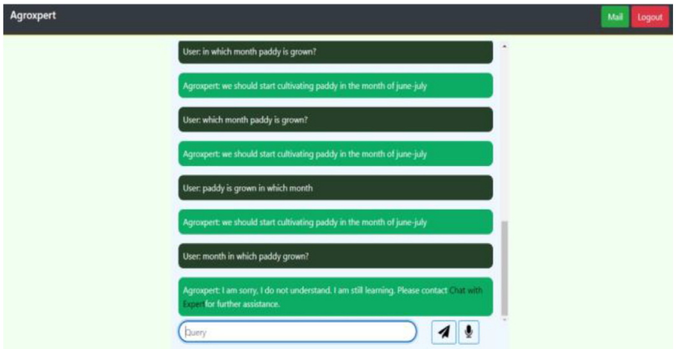


Fig. 5. Farmer sign in.



User: in which month paddy is grown?

Agroxpert: we should start cultivating paddy in the month of june-july

User: which month paddy is grown?

Agroxpert: we should start cultivating paddy in the month of june-july

User: paddy is grown in which month

Agroxpert: we should start cultivating paddy in the month of june-july

User: month in which paddy grown?

Agroxpert: I am sorry, I do not understand. I am still learning. Please contact Chat with Expert for further assistance.

Fig. 6. Farmer chatbot page.

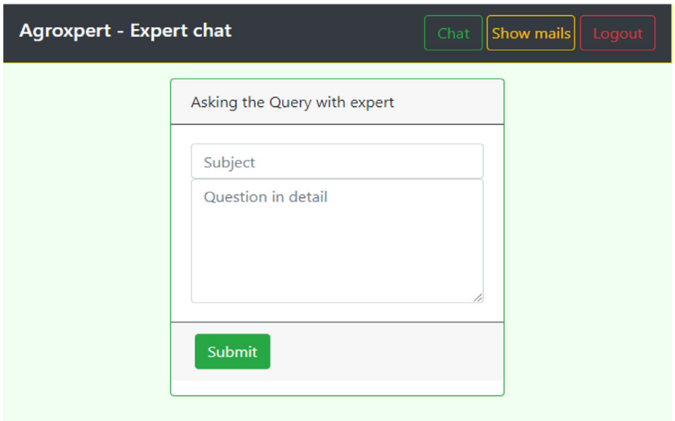
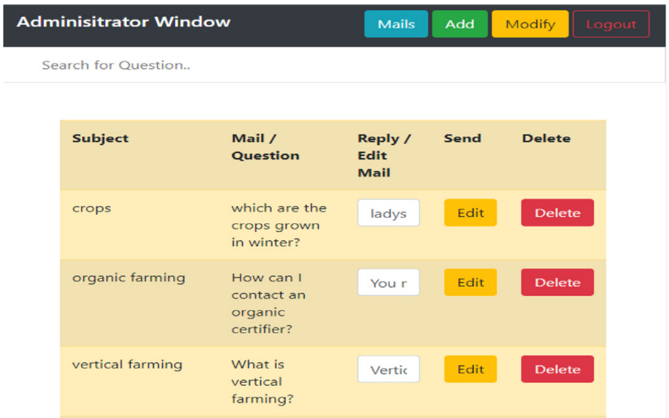


Fig. 7. Expert chat.



Subject	Mail / Question	Reply / Edit Mail	Send	Delete
crops	which are the crops grown in winter?	ladys	Edit	Delete
organic farming	How can I contact an organic certifier?	You r	Edit	Delete
vertical farming	What is vertical farming?	Vertic	Edit	Delete

Fig. 8. Mail details shown in the admin side.

similar login and registration page the only difference is that the admin needs to have the email ID instead of the phone number.

Once the farmer logs in successfully, he will be directed to the chatbot page, where he can ask questions via text or voice. Once the farmer types the question he can send the question using the send button located as shown in Fig. 6. The microphone button represents the input of the farmer's query through the voice input for which the browser asks permission for accessing the microphone. The voice input is automatically converted to the text [14–16].

Fig. 7 represents the feature of mailing the expert for additional information regarding the concerned topic.

Fig. 8 gives main details in the admin side and it looks similar to the inbox in Gmail where you can see all the details of mail. Admin can edit and delete the reply he has sent [17–19].

Expert farmer (Admin) can add new question and answer for training and improving the chatbot. He can even provide the URL of some valid site for the farmer's reference (Fig. 9).

Fig. 10 represents Modify page where existing questions could be edited or deleted.

The pattern matching method is implemented in the chatbot. The chatbot will be trained with the questions and the pattern matching is

Table 3
Test result.

Input	Expected result	Response from chatbot	Accept or reject
In which month paddy is grown?	we should start cultivating paddy in the month of june-july	we should start cultivating paddy in the month of june-july	Accepted
Which month paddy is grown?	we should start cultivating paddy in the month of june-july	we should start cultivating paddy in the month of june-july	Accepted
paddy is grown in which month	we should start cultivating paddy in the month of june-july	we should start cultivating paddy in the month of june-july	Accepted
month in which paddy is grown?	we should start cultivating paddy in the month of june-july	Sorry, still learning, talk with the expert.	Rejected

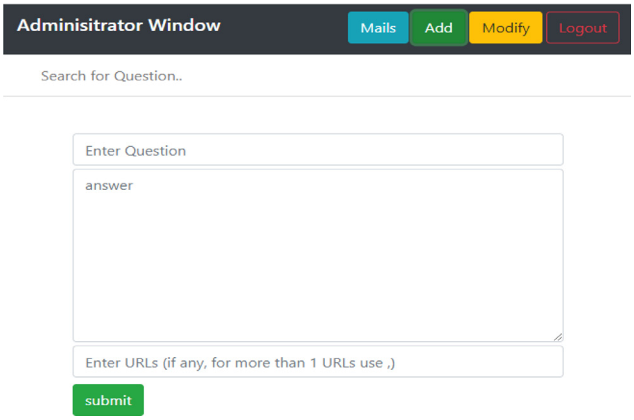
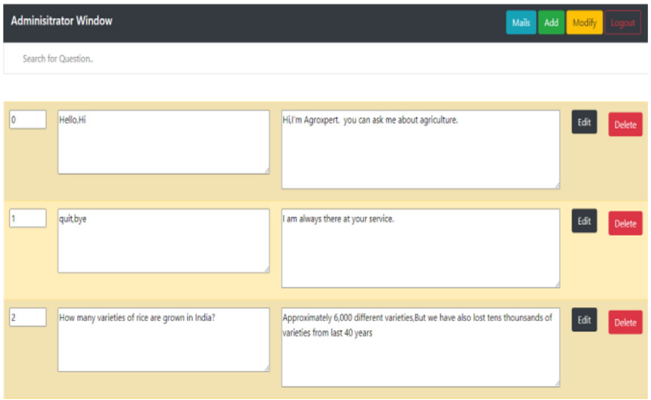


Fig. 9. Adding new query-response in the admin side.



0 — Hello,Hi — Hi,I'm Agroxpert. you can ask me about agriculture.

1 — quit,bye — I am always there at your service.

2 — How many varieties of rice are grown in India? — Approximately 6,000 different varieties,But we have also lost tens thounsands of varieties from last 40 years

Fig. 10. Editing the existing query-response in the admin side.

done between the input query and the data used during training. Each question must have a pattern, and that pattern is unique from other questions. The pattern for a particular question can be keywords. Even if the query string's order is changed or jumbled, required result is obtained. The chatterbot uses the best match model and threshold similarity index can be set to the best match. If the query is exactly like the one which is present in the dataset, then the answer obtained is 100% accurate. But if the question is not exactly as it is in the dataset then getting the answer accurate depends on the threshold similarity index and best match algorithm [20,21]. It is 90% accurate to get the required answer if the sentence is not exactly as in the dataset. Table 3 depicts the result that is obtained after reforming the sentence "In which month paddy is grown".

Ideally, all questions should be answered by the chatbot. Unfortunately, the chatbot is sometimes unaware of this problem. In this case, if the low-confidence adapter is not configured, the default response can be returned as a response after the search. The low confidence threshold is configurable. In the example, the threshold is set to 0.75. This means that if the input match rate is less than 75%, a standard assertion will be returned as a response. Even if it is not perfect, the chatbot is limited to the agriculture theme. The chatbot can quickly provide real-time mailing features which involves talking to experts and clearing the doubts. For the question, the chatbot can successfully answer questions that match the training data set.

Table 4
Accuracy comparison with previous works.

Reference	Algorithm/ Method /Techniques Used	Accuracy (%)
Sawant et al. [1]	KNNDecision treeRandom forest	72.5%75%78%
Xie et al. [19]	Best Match with Confidence Stopper	75%
Arora et al. [4]	CNN Architecture	94%
Proposed method	Chatterbot	96%

The Accuracy obtained for algorithms or methods shows comparative analysis of prediction accuracy with that of existing methods. It is very clear from the table that the accuracy of the proposed method is more from other existing methods or techniques (Table 4).

5. Conclusion and future work

Agriculture is the backbone of the economy. It provides food, raw materials and even employment opportunities to so many people. It's been practiced for over thousands of years now. Over the years, due to the advancement and latest trends in technology, agricultural techniques have also evolved. But in India, there are small and marginal farmers who practice old, traditional methods due to lack of resources. Hence, this system works as a virtual assistant which guides the farmers by answering all their queries regarding agricultural practices, thereby helping them generate higher profits. In future the chat may be enabled to send queries in the form of pictures and videos and get the right response. Future scope would be enabling audio and video calling features with the expert, enabling video and image response from the chatbot and language translation i.e. from English to the local language that is understood by the farmers. Overall, this initiative of implementing chatbots will assist farmers and thereby boost the economy of the country.

Abbreviations

ML -	machine learning
NLP -	natural language processing
CNN -	convolutional neural network
AI -	artificial intelligence
IOT -	internet of things
HTML -	hypertext markup language
CSS -	cascading style sheets
AJAX -	asynchronous javascript and XML
CSV -	comma-separated values
SQL -	structured query language

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] D. Sawant, A. Jaiswal, J. Singh, P. Shah, Agribot - an intelligent interactive interface to assist farmers in agricultural activities, in: Proceedings of the Bombay Section Signature Conference (IBSSC), IEEE, 2019.

[2] P.Y. Niranjan, V.S. Rajpurohit, R. Malgi, A survey on chat-bot system for agriculture domain, in: Proceedings of the 1st International Conference on Advances in Information Technology (ICAIT), 2019.

[3] F. Patel, R. Thakore, I. Nandwani, S.K. Bharti, Combating depression in students using an intelligent Chatbot: a cognitive behavioral therapy, in: Proceedings of the 16th India Council International Conference (INDICON), IEEE, 2019.

[4] B. Arora, D.S. Chaudhary, M. Satsangi, M. Yadav, L. Singh, P.S. Sudhish, Agribot: a natural language generative neural networks engine for agricultural applications, in: Proceedings of the International Conference on Contemporary Computing and Applications (IC3A), 2020.

[5] R.B. Mathew, S. Varghese, S.E. Joy, S.S. Alex, Chatbot for disease prediction and treatment recommendation using machine learning, in: Proceedings of the 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019.

[6] E. Kasthuri, S. Balaji, A chatbot for changing lifestyle in education, in: Proceedings of the 3rd International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), 2021.

[7] M.M. Rahman, R. Amin, M.N. Khan Liton, N. Hossain, Disha: an implementation of machine learning based Bangla healthcare chatbot, in: Proceedings of the 22nd International Conference on Computer and Information Technology (ICCIT), 2019.

[8] N.N. Khin, K.M. Soe, University chatbot using artificial intelligence markup language, in: Proceedings of the Conference on Computer Applications(ICCA), IEEE, 2020.

[9] H. Koundinya K., A.K. Palakurthi, V. Putnala, A. Kumar, K. Smart, College chatbot using ML and python, in: Proceedings of the International Conference on System, Computation, Automation and Networking (ICSCAN), 2020.

[10] G.K. Vamsi, A. Rasool, G. Hajela, Chatbot: a deep neural network based human to machine conversation model, in: Proceedings of the 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2020.

[11] N. Bhartiya, N. Jangid, S. Jannu, P. Shukla, R. Chapaneri, Artificial neural network based university chatbot system, in: Proceedings of the Bombay Section Signature Conference (IBSSC), IEEE, 2019.

[12] U. Kiruthika, S.K.S. Raja, V. Balaji, C.J. Raman, E-agriculture for direct marketing of food crops using chatbots, in: Proceedings of the International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), 2020.

[13] R. Gunawan, I. Taufik, E. Mulyana, et al., Chatbot application on internet of things (IoT) to support smart urban agriculture, in: Proceedings of the 5th International Conference on Wireless and Telematics (ICWT), IEEE, 2019.

[14] T.N. Nguyen, B.H. Liu, S.I. Chu, D.T. Do, T.D. Nguyen, WRSNs: toward an efficient scheduling for mobile chargers, IEEE Sens. J. 20 (12) (2020) 6753–6761.

[15] N.T. Le, J.W. Wang, C.C. Wang, T.N. Nguyen, Automatic defect inspection for coated eyeglass based on symmetrized energy analysis of color channels, Symmetry 11 (12) (2019) 1518.

[16] V. Hristidis, Chatbot technologies and challenges, in: Proceedings of the First International Conference on Artificial Intelligence for Industries (AI4I), 2018.

[17] D.L. Vu, T.K. Nguyen, T.V. Nguyen, T.N. Nguyen, F. Massacci, P.H. Phung, HIT4Mal: hybrid image transformation for malware classification, Trans. Emerg. Telecommun. Technol. 31 (11) (2020) e3789.

[18] B.D. Parameshachari, H.T. Panduranga, S. liberata Ullo, Analysis and computation of encryption technique to enhance security of medical images, IOP Conf. Ser. Mater. Sci. Eng. 925 (1) (2020) 012028 (VolNo.IOP Publishing.

[19] Q. Xie, et al., Chatbot application on cryptocurrency, in: Proceedings of the Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr), IEEE, 2019.

[20] G.B. Rajendran, U.M. Kumarasamy, C. Zarro, P.B. Divakarachari, S.L. Ullo, Land-use and land-cover classification using a human group-based particle swarm optimization algorithm with an LSTM Classifier on hybrid pre-processing remote-sensing images, Remote Sens. 12 (24) (2020) 4135.

[21] B.D. Parameshachari, H.T. Panduranga, Secure transfer of images using pixel-level and bit-level permutation based on knight tour path scan pattern and henon map, in: Cognitive Informatics and Soft Computing, Springer, Singapore, 2021, pp. 271–283.