Agri Chatbot for Farmers

*Narayana Darapaneni Director - AIML Great Learning/Northwestern University Illinois, USA darapaneni@gmail.com*

*Divya Abraham Student – Great Learning Bangalore, India*

*dvyjodd@gmail.com*

*Sudha B. G Centre Head Great Learning Bangalore,India.*

*sudha.bg@greatlearning.in*

*Swagatika Samal Student – Great Learning Bangalore, India*

*swagatikasamal16@gmail.com*

*Aslam Pasha Student - Great Learning Bangalore, India aslampasha199510@gmail.com*

*Vishnu Bansal Student – Great Learning Bangalore, India*

*vishnukbansal@yahoo.com*

*Abstract*—Agriculture is the backbone of India and a significant part of the Indian economy, and the country now has one of the highest rates of farm producers in the world. Farmers lose production because they are unaware of new technology and parameters that might assist them boost their produce. Farmers need hand holding with support of technology. The Government of India has made public all call records of the KCC (Kisan Call Center) from January 2015 to September 2017. Similar inquiries to those revealed in the KCC dataset were given to the researchers by farmers and agribusiness professionals. Based on both sources, we identified four important areas in need of information support: plant protection, pests (diseases), weather, and best practices. Deep Learning and Natural Language ChatBot applications have grown fast in recent years. They are utilized in a variety of sectors, including customer service, reservation systems, and as personal assistants. Enterprises use ChatBots to provide better and more efficient service to their clients. Even with such technical innovation, professional guidance does not reach farmers on time. Farmers are still heavily reliant on their peers' wisdom to solve difficulties in their fields. These technologies have not been used properly to provide farmers with the necessary information in a timely way. The goal of this project is to create "AgriBot," a closed domain ChatBot for the sector of agriculture. Farmers may communicate with closed-domain and receive professional guidance in their sector. The RASA Open-Source Framework serves as the foundation for AgriBot. The AgriBot recognizes the purpose and entity from the user's utterances and retrieves and shares the remedy from the database. We tested the Bot with existing data and found it to be promising.

Keywords—Rasa, Deep Learning, NLP

# Introduction (*Heading 1*)

Agriculture is one of India's most important sectors. In 1983, the agricultural sector employed over 77% of the workforce and contributed 34% of GDP. These figures have decreased over time, and they currently employ just 40% of the Indian workforce, and their contribution to GDP has also decreased. Many factors contribute to the such drop in the importance of Agriculture. One of the primary factors is a declining return on investment in agriculture. The ROI can be improved if farmers have sufficient access to expert advice on how to protect their crops, where can they get their seeds, what crops to farm in which season, etc. The Indian government established the Kisan Call Center to offer such information to farmers. The call centers are available in 22 languages, and professionals are on hand to answer Farmers' questions. The call centre is divided into two tiers. Level 1 responds to the call, gathers basic information from the user, and attempts to give the necessary information. When Level 1 is unable to resolve the uncertainty, the request will be passed to Level 2 - Subject Matter Expert with the necessary knowledge. The SME has 72 hours to respond with the necessary details. Every year, the KCC gets between 50 and 60 lakh calls. When demand increases, call centre resources cannot be expanded forever. If Level 1 cannot answer the queries, the customer may have to wait 72 hours to receive the necessary information. Furthermore, while employing specialists, the government looks for persons who are educated about the crops that grow in that area. . All of these problems may be solved by developing a ChatBot that holds the expertise of all of the region's agricultural professionals. Unlike a contact centre, a ChatBot can handle an increase in demand or a rush in calls. The consumer does not have to wait for the expert to respond within 72 hours. To test this, we created a Agribot (Chatbot) using the Rasa Framework. The Agribot was taught in areas such as crop protection and nutrient management on some hand-picked crops.

# Ease of Use

## Literature Survey

The discussion about chatbots cannot happen without the mention of Alan Turing, who is widely considered as Father of Artificial Intelligence. He is the one who proposed Turing Test and started the idea of “Can Machines think?”. Basically, the idea is to build a machine that can mimic a human and make another person believe that he is actually conversing with a human. This popularized the idea of Chatbot. Then in 1966, the first Chatbot ELIZA was developed by MIT professor Joseph Weizenbaum. ELIZA identified the keywords from the user’s input and applied a set of decomposition and reassembly rules and generated a response to the user. This Chatbot is based on Pattern Recognition Algorithm and the response was repetitive.

NOW, WHAT IS A PATTERN RECOGNITION ALGORITHM? Pattern recognition is a data analysis method that uses machine learning algorithms to automatically recognize patterns and regularities in data. This data can be anything from text and images to sounds or other definable qualities. Pattern recognition systems can recognize familiar patterns quickly and accurately.

In 1995, the Chatbot A.L.I.C.E was developed using pattern matching and AIML – Artificial Intelligence Markup Language. This won the Loebner Prize as “the most human computer” at the annual Turing Test contests in 2000 and 2001. Many considered A.L.I.C.E as an extension of ELIZA. But the major difference is in the categories of Knowledge. A.L.I.C.E had more than 40000 categories of knowledge, whereas the original ELIZA had only 200 categories. In the 1990s, a lot of research in this area led to the development of many conversational systems. But they were designed mainly to specialize in a specific domain.

The best example is DARPA Airline Travel Information System (ATIS), which was designed to book Airline tickets.

The Chatbot till then was mostly based on pattern-matching algorithms or on markup language like AIML. Chatbot technology took a big stride after the rise of Deep Learning. In this approach, the Chatbot/Neural network was trained on large amount of data, which made it possible for the Bot to generate a reply to the user’s utterance. A chatbot is a kind of application in which the output of the Model at time "t" is dependent on all the previous input that it received till the time "t-1". The basic DNN model does not have any memory to remember its previous input and so it cannot be used for Chatbot.

Below are the few examples of chatbot developments in the field of agriculture:

Virtual Conversational Assistant: This paper presents a query about agriculture, gets the response in text. The future enhancement can be done by giving the response in their regional language itself and prediction can be extended to rainfall and productivity. The usage of cognitive technologies predicts exciting times ahead for agriculture on its road towards efficiency, sustainability and meeting the world’s food needs. This conversational assistant uses Natural Language Processing techniques to understand the user queries in their natural language. This will make the system understand even the grammatically not well-defined sentences as input queries. The user queries undergo the pre-processing stage where the query is first tokenized into words, then the stop words like, a, is, the, etc., are removed so that it would not contribute to the probability of classifying the queries based on their respective classes and then the stemming process is carried out where the words are converted to their root words. The words are converted to a bag of words and then converted to a vector form so that they can be processed efficiently by the classification algorithm. The bot is then trained with the training dataset. Based on the training set data, a neural network is constructed and error is optimized using the gradient descent algorithm. The test data set undergoes the same pre-processing stages, classification and neural network construction. The class with the highest probability is iterated to get the accurate results.

Farm Chat: The analysts at Farm Chat created an information base for potato cultivation using the KCC dataset, and also collected data from developmental interviews with smallholder ranchers and agri-experts. For each of the distinguished subjects, they inquired the two agri-experts and applied illustrations of commonplace farmer questions, the follow-up questions that they would inquire in arrange to get at the issue, and the ultimate resolution they would provide. All such conversations were included to the CSV file, and the informational advice was included within the Farm Chat information base.

## Materials and methods

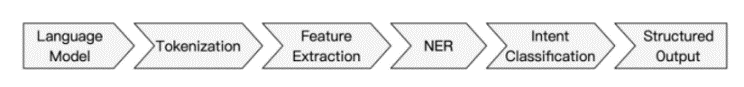
We propose a solution of a chatbot that is interactive for the farmers and easy to answer their queries in an effective manner. The Workflow starts with the hello message reply by the bot and queries will be asked by farmers in which the possible solution will be listed out based on the trained dataset we give for training. We use RASA as a tool to build custom AI chatbots using Python and natural language understanding (NLU). It also allows the user to train the model and add custom actions.

Rasa architecture contains two main parts :

**RASA NLU**: RASA NLU is an open-source natural language processing tool U is responsible for intent recognition and entity extraction. It uses supervised learning algorithms to fulfill this function. A proper number of examples including intent and entity information are needed for training the NLU model. It has a very flexible software architecture design and supports various kinds of algorithms. The implementations of those algorithms are called COMPONENTS.

Rasa NLU introduces pipelines as the component configuration system to achieve this. CORE ELEMENTS OF Rasa NLU: The format of NLU training data. – Rasa NLU components. – Configuring our Rasa NLU via a pipeline. – The output of Rasa NLU – Training and serving Rasa NLU – Practice - building the NLU part of an Agri Bot.

Overview of Rasa NLU Components:



• Language model component: This loads the language model files to support the following components. There is multiple inbuilt language models available and we have used Language Model Featurizer in Agri Chatbot.

• Tokenizer component: Tokenizer components are closely related to languages. No tokenizer can support all languages. You should choose the appropriate tokenizer according to your target language. We have supported the English language hence Whitespace Tokenizer is used for Agri Chatbot.

• Featurizer component: For both entity extraction and intent classification, features provided from upstream components are required. Developers can use multiple components to do feature extraction. We have used Regex Featurizer in Agri Chatbot. Entity extraction component: Rasa supports multiple entity extraction components. Rasa recommends DIET Classifier because it usually has better performance hence, we have gone through with the recommendation.

• Intent classifier component: This component is used to classify the intent from each query. Rasa recommends DIET Classifier for better performance.

• Structure output: This organizes the prediction results into structured data and outputs it. This part is not a component but a built-in function within the pipeline. Developers are not able to directly access it as a component.

**RASA CORE:** Rasa Core it is the dialogue management part of Rasa. In Rasa Core, these functions have all been integrated, and users can use Rasa’s dialogue management functions in an end-to-end machine learning-based manner.

**SESSION CONFIGURATION:** The session is a conversation between the user and the bot. One session can persist for multiple dialogue turns. Currently,  
Rasa supports two types of session configurations:  
• session\_expiration\_time: defines the expiration time (in minutes) after the user gets the newest message. There is no  
expiration time if it is set to 0.  
• carry\_over\_slots\_to\_new\_session : defines whether the system should bring the slots from the previous session into  
the new session. If set to false, the new session will not get the slot values from the previous session.

TRAINING DATA PREPARATION: Here we will show you the part of the training file:

* nlu.yml: In the NLU training data file, we have training data for multiple intents. We can easily infer the meaning from the intent’s name and the entity’s name. In our project, all the training data is stored in the data/nlu.yml file .
* .
* stories.yml: Rasa learns from conversations and manages knowledge by training on stories. The story is a high-level semantic way of recording conversations. It records not only the expressions from users but also the correct state change within the system.
* domain.yml: A domain defines all the information a chatbot needs to know, including intents, entities, slots, actions, forms, and responses. All this information gives clear definitions of the inputs and outputs of a model.

Graphical user interface, text, application

Description automatically generated

TRAIN THE MODEL: Once we prepare all the training data and place the file or respective folder structure, we can start the model to adjust the weights based on the train data. We can prepare the testing data as well to check the module’s accuracy.

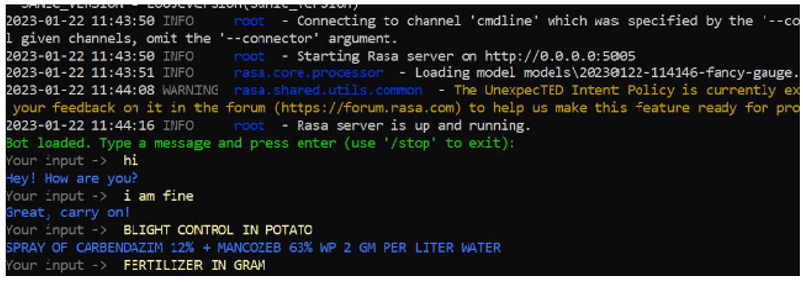
Following are the commands to create and execute the interactive shell.

**Rasa train**: This command will use the NLU train data, story data, and configuration to train the model. This will generate  
the model zip file and will be stored default model folder.

**Rasa shell**: This command will run the rasa server and open the interactive command line interface to interact with  
bot.

## Result :

## AGRI ChatBot was able to successfully converse with the user and identify the intent and entity for which it was trained. The Bot retrieved appropriate response from the database and shared it with the user. It worked well even for questions which are out of scope for the Bot’s functionality. The AGRI ChatBot is designed to seamlessly answer queries of the beneficiaries and is an attempt to provide the mass farmers a communication channel through which they can ask their queries and get resolution at any time without needing to worry about call center timings and network congestion issues. Also since Kissan website has one of the leading chat mediums in India, with a mass reach we choose to integrate the chat-bot to this platform. The bot will respond to greeting by the user with an appropriate greeting message of introduction about itself, and then the user can proceed with their queries. After answering the query, the bot will check for the satisfaction of the answer, if satisfied, it will welcome user for any further query. If answer is not satisfactory, the bot will provide the details of the call center to the farmer to contact on phone during the working hours. The RASA platform on which the bot has been built provides a number of ways to integrate the NLP algorithms to build out a solution. We can use the built in models or feed any custom model to the platform and build the final model via transfer learning using the data provided to it.



The above figures show the response from the bot for scenarios where the confidence rate was high enough to give the answer to the user. In response to the FAQ (frequently asked questions), a beneficiary may also ask some irrelevant  
queries, which the bot may not be able to answer. In that case, the bot will go for a Fallback response since the confidence level for answering the query will be less than 70 percent.

We have tested the built model and found it to be satisfactory in providing a relevant response.

# Prepare Your Paper Before Styling

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections A-D below for more information on proofreading, spelling and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

## Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

## Units

* Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as “3.5-inch disk drive”.
* Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
* Do not mix complete spellings and abbreviations of units: “Wb/m2” or “webers per square meter”, not “webers/m2”. Spell out units when they appear in text: “. . . a few henries”, not “. . . a few H”.

Identify applicable funding agency here. If none, delete this text box.

* Use a zero before decimal points: “0.25”, not “.25”. Use “cm3”, not “cc”. (*bullet list*)

## Equations

The equations are an exception to the prescribed specifications of this template. You will need to determine whether or not your equation should be typed using either the Times New Roman or the Symbol font (please no other font). To create multileveled equations, it may be necessary to treat the equation as a graphic and insert it into the text after your paper is styled.

Number equations consecutively. Equation numbers, within parentheses, are to position flush right, as in (1), using a right tab stop. To make your equations more compact, you may use the solidus ( / ), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in:

*a**b* 

Note that the equation is centered using a center tab stop. Be sure that the symbols in your equation have been defined before or immediately following the equation. Use “(1)”, not “Eq. (1)” or “equation (1)”, except at the beginning of a sentence: “Equation (1) is . . .”

## Some Common Mistakes

* The word “data” is plural, not singular.
* The subscript for the permeability of vacuum **0, and other common scientific constants, is zero with subscript formatting, not a lowercase letter “o”.
* In American English, commas, semicolons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
* A graph within a graph is an “inset”, not an “insert”. The word alternatively is preferred to the word “alternately” (unless you really mean something that alternates).
* Do not use the word “essentially” to mean “approximately” or “effectively”.
* In your paper title, if the words “that uses” can accurately replace the word “using”, capitalize the “u”; if not, keep using lower-cased.
* Be aware of the different meanings of the homophones “affect” and “effect”, “complement” and “compliment”, “discreet” and “discrete”, “principal” and “principle”.
* Do not confuse “imply” and “infer”.
* The prefix “non” is not a word; it should be joined to the word it modifies, usually without a hyphen.
* There is no period after the “et” in the Latin abbreviation “et al.”.
* The abbreviation “i.e.” means “that is”, and the abbreviation “e.g.” means “for example”.

An excellent style manual for science writers is [7].

# Using the Template

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

## Authors and Affiliations

**The template is designed for, but not limited to, six authors.** A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

### For papers with more than six authors: Add author names horizontally, moving to a third row if needed for more than 8 authors.

### For papers with less than six authors: To change the default, adjust the template as follows.

#### Selection: Highlight all author and affiliation lines.

#### Change number of columns: Select the Columns icon from the MS Word Standard toolbar and then select the correct number of columns from the selection palette.

#### Deletion: Delete the author and affiliation lines for the extra authors.

## Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is “Heading 5”. Use “figure caption” for your Figure captions, and “table head” for your table title. Run-in heads, such as “Abstract”, will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named “Heading 1”, “Heading 2”, “Heading 3”, and “Heading 4” are prescribed.

## Figures and Tables

#### Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation “Fig. 1”, even at the beginning of a sentence.

1. Table Type Styles

| Table Head | Table Column Head | | |
| --- | --- | --- | --- |
| Table column subhead | Subhead | Subhead |
| copy | More table copya |  |  |

1. Sample of a Table footnote. (*Table footnote*)
2. Example of a figure caption. (*figure caption*)

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity “Magnetization”, or “Magnetization, M”, not just “M”. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write “Magnetization (A/m)” or “Magnetization {A[m(1)]}”, not just “A/m”. Do not label axes with a ratio of quantities and units. For example, write “Temperature (K)”, not “Temperature/K”.

##### Acknowledgment *(Heading 5)*

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

##### References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.