Text Based Smart Answering System in Agriculture using RNN

Raji Sukumar A[1], **Hemalatha N**[2], **Sarin S**[1], and
Rose Mary C A[2]

[1]Techtern Pvt Ltd Kannur. Kerala,
[2]Aloysius Institute of Management and IT (AIMIT),Mangalore University
rajivinod.a@gmail.com, hemalatha@staloysius.ac.in,
sarinsukumar@gmail.com, rosemaryca@gmail.com

Abstract

Agriculture is an important aspect of India's economy, and the country currently has one of the highest rates of farm producers in the world. Farmers need hand holding with support of technology. A chatbot is a tool or assistant that you may communicate with via instant messages. The goal of this project is to create a Chatbot that uses Natural Language Processing with a Deep Learning model. In this project we have tried implementing Multi-Layer Perceptron model and Recurrent Neural Network models on the dataset. The accuracy given by RNN was 97.83%.

1 Introduction

Agriculture contributes around 16 percent of India's GDP and employs about 52 percent of the country's population, making it a significant part of the country's economic growth. According to the Farmers' Portal, agriculture's rapid expansion is necessary not just for self-sufficiency but also for earning vital foreign exchange. One of the reasons for this is that individuals in the farming industry are relatively sluggish to accept emerging innovations. Field officers have traditionally visited farmlands to give training, guidance, and assistance to farmers. The data demonstrates that mobile connection is increasing at an exponential rate, which helps IT services promote agricultural information. The government is having difficulty disseminating important agricultural information. Furthermore, the difficulties are exacerbated by the dissemination of disinformation. These issues exist as a result of the huge linguistic variety and the rural population's lack of trust in contemporary technologies. In such a situation, using mobile devices to disseminate agricultural information looks to be a viable option [1] (K., 2020). Chat-Bot sys-

---

[1]https://en.wikipedia.org/wiki/Agriculture_in_India/

tems are a type of natural language processing that demands the system to be taught in human language in order to meet the user's demands. Agriculture is the most important sector for a country's development. Farmers are now unaware of the most modern technology and methods employed in agriculture. The challenge of extracting meaningful answers using machine learning techniques has been researched by numerous machine learning specialists, and sophisticated machine learning approaches have been created. These methods are used to obtain the correct answer. We may name this an Agriculture Question Answering System, since the farmer can ask the system a question, and the system will answer (Heller et al., 2005; Beaudry et al., 2019; Sutoyo et al., 2019). With the advancement of technology, farmers must study and address the challenges. As a result, the goal is to create a chatbot system that delivers accurate responses to queries. According to a major study in the field of chatbot systems, there is no agriculture-specific system that can provide precise and rapid answers to farmers' questions. To solve this issue, the suggested system uses the RNN (Recurrent Neural Network) deep learning method to offer accurate responses to the queries asked.

2 Problem Statement

The traceability software from Source Trace gives you complete visibility into the agricultural value chain. It has an influence on farmers' lifestyles, helps an organization adopt data-driven agriculture, and fosters trust and improved interaction with stakeholders. Agriculture employed half of India's workforce and provided 17–18% of the country's GDP. Agriculture and related industries such as animal husbandry, forestry, and fisheries accounted for 15.4% of GDP in 2016 and employed around 31% of the workforce in 2014. The

Figure 1: Dataset format

goal of this project is to create a Chatbot that uses natural language processing to facilitate remote interaction between users/farmers and the agriculture environment. We aim to create a chatbot that can answer basic questions from farmers and give possible agricultural knowledge and solutions. Because this chatbot has been educated in natural language processing, it can learn on its own and improvise responses. The study's target audience is agriculturists or farmers. Their work will be based on the model established. As a consequence, agriculturists will be able to benefit from the study's findings. It has an influence on farmers' livelihoods, helps organizations adopt data-driven agriculture, and fosters trust and improved interaction with stakeholders. The farmers' favorable reaction suggests that conversational intelligence, as a technology supplied via the omnipresent smartphone, can be a useful tool for improving information access in rural areas for those with low literacy and technological expertise.

3 Methodology

3.1 Dataset source and format

The dataset used for this project is taken from data world repository. A multidisciplinary or specialized conversational chatbot is possible. The chatbot's capabilities are influenced by the amount of data utilised to train it. The data is saved as a json file. Tags, patterns, responses, and context are used to organise the data: (.A and Anto, 2013)

Tags: Possible classes of user intention for asking a question.

Patterns: The ways in which users usually ask questions relating to a particular tag.

Responses: Predefined responses for each tag in the dataset from which the model can choose to respond to a particular question.

Context: Contextual words relating to a tag for easy and better classification of what the user intends with their request.

3.2 Workflow diagram

Businesses must understand the workflow of these bots in order to build a chatbot that offers appealing outcomes(Weng, 2019). From the time the chatbot receives a user's query until the time it delivers an answer, the data travels through a number of algorithms that assist the chatbot in comprehending the input. Specifically, most chatbots use several categorization techniques to build up their fundamental architecture Figure 2.

3.3 Data preprocessing

The data has been extracted in json format itself using google colab. Using json.loads() the data is loaded into the system. Google Research's Collaboratory, or "Colab" for short, is a product. Colab is a web-based Python editor that allows anybody to create and run arbitrary Python code. It's notably useful for machine learning, data analysis,
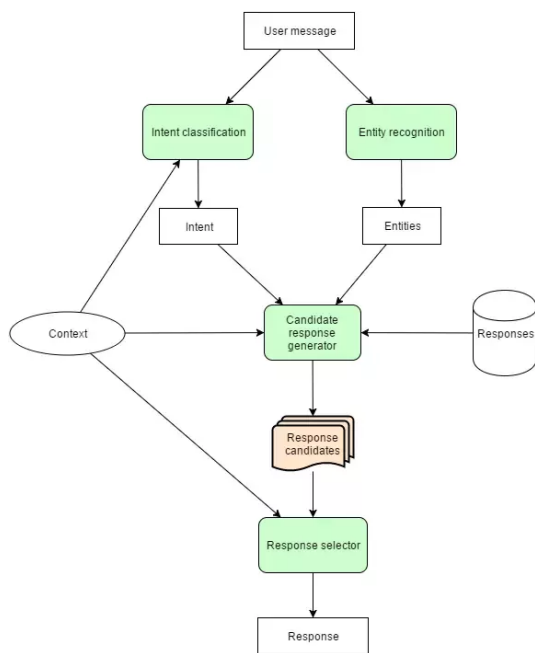
Figure 2: Chatbot Architecture

and teaching. On Google Colab I went with CPU runtime in the first notebook and with the GPU runtime in the second. Then the runtime has been changed to GPU to process the codes faster. We can't just fit a machine learning or deep learning model to the raw text. To begin, we must divide words, handle punctuation and cases, and more in order to prepare the data for modelling. In NLP, cleaning up text data is job specific. For this conversational chatbot we're building, we can do the following. With our intents JSON file loaded, we can now begin to organize our documents, words and classification classes (.A and Anto, 2016).

- Tokenization

- Stemming

- Lemmatization

- Removal of stop words

- Spelling correction

- Normalization

- Removal of punctuation marks

- Creation of training data

1)Tokenization: Tokenization is the process of breaking down a text corpus into constituent words, such as breaking down a phrase, sentence, paragraph, or even an entire text document into smaller units like individual words or terms. A token is the name given to each of these smaller components. Tokenization can be done manually using white space splitting or using specific tools in libraries like NLTK. After tokenization we organized the dataset into words, classes and documents list.

2) Word stemming : The process of creating morphological variations of a root/base word is known as stemming. Stemming algorithms or stemmers are terms used to describe stemming programmes. The terms "chocolates," "chocolatey," and "choco" are reduced to the root word "chocolate," while "retrieval," "retrieved," and "retrieves" are reduced to the stem "retrieve." NLTK has LancasterStemmer class with the help of which we can easily implement Lancaster Stemmer algorithms for the word we want to stem.

3) Lemmatization : Lemmatization is the act of combining a word's several inflected forms into a single item that can be examined. Lemmatization is similar to stemming, except it gives the words context. As a result, it connects words that have similar meanings to one another. The words are morphologically analysed during lemmatization.

4)Removal of stop words: Stop word removal is supported by NLTK, and the list of stop words may be found in the corpus module. To eliminate stop words from a phrase, break your text into words and then check to see if the word is in the NLTK list of stop words.

5) Spelling correction: It is the process of correcting a word's spelling. Because brute force comparisons are extremely time intensive, most spell correction algorithms employ min-edit functions. We need to utilise word lengthening first in order for min-edit features to operate properly. As a result, word lengthening affects our spell correction. Although NLTK lacks a spell-checking module, there are several libraries that can accomplish this function. For this, I'll be utilising the pyspellchecker module. Pyspellchecker is a library for assessing if a word is misspelt and, depending on word frequency, what the likely right spelling is.

6) Text case conversion: We'll use this approach to change the words to lower case or upper case. Although sometimes ignored, one of the simplest and most efficient forms of text preparation is to lowercase all of your text data. It can be used to solve most text mining and NLP issues, and it may
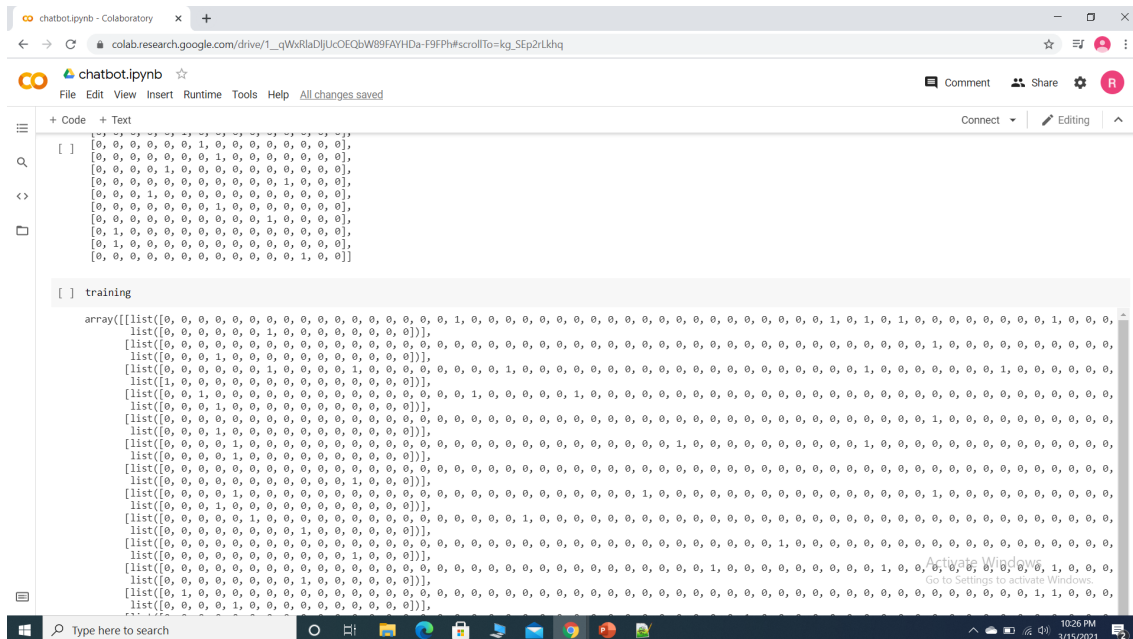
Figure 3: Data after transformation (bag of words)

be very useful when your dataset isn't very huge. It also greatly improves anticipated output consistency.

7) Removal of punctuation marks: Noise reduction is the process of eliminating letters, numbers, and text fragments that might obstruct your text analysis. One of the most important text preparation procedures is noise reduction. It's also quite domain-specific. The framework's replacement duties continue with noise reduction. While the first two major steps of our framework (tokenization and normalisation) could be applied to almost any text chunk or project as-is (barring the decision of which exact implementation to use, or skipping certain optional steps, such as sparse term removal, which does not apply to every project), noise removal is a much more task-specific section of the framework.

8) Creation of training data: Bag of Words (BOW) is a vector space representational model for unstructured text that is one of the simplest. A vector space model is a mathematical model for representing unstructured text (or any other data) as numeric vectors, with each dimension of the vector corresponding to a distinct feature property. Each text document is represented as a numeric vector in the bag of words model, with each dimension being a single word from the corpus and the value being its frequency in the document, occurrence (denoted by 1 or 0), or even weighted values. The term comes from the fact that each text is repre-

sented as a 'bag' of its own words, with no concern for word ordering, sequences, or grammar (Jiao, 2020; Bhagwat, 2018).

We need to translate the words into bags of words with arrays containing 0/1. The array length will be equal to vocabulary size, and 1 will be set when a word from the current pattern is located in the given position. Training data — X (pattern converted into array [0,1,0,1..., 0]), Y (intents converted into array [1, 0, 0, 0,...,0], there will be single 1 for intents array).The transformed data is as follows in Figure 3.

3.4 Model implementation

The processed data is used for intent classification using the models so that the model gives promising results. The data is modelled using two deep neural network models that is Multi-Layer Perceptron and Recurrent Neural network model.

1) Multilayer Perceptron (MLP)

A feedforward network having one input layer, one output layer, and at least one hidden layer is known as a multilayer perceptron. Non-linear activation functions, such as the hyperbolic tangent or logistic function, are used to categorize data that is not linear in nature. Every node in the current layer is connected to every node in the following layer, making the network fully connected. After that, the layers are fully connected in a chronological order (input to output), which is known as feedforward: input -> hidden -> output. Tensorflow's

666

tflearn framework is used to build the MLP network. The fundamental neural network we'll be utilizing is called a Multilayer Perceptron (MLP for short) (Weng, 2019), and it's shown in Figure 4.
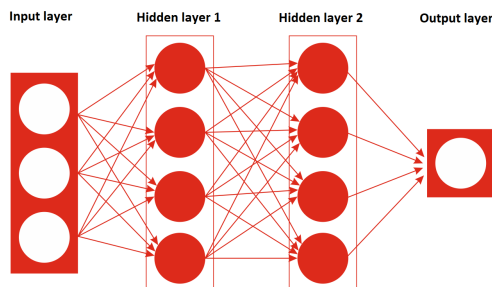

Figure 4: Neural Network – Multilayer Perceptron (MLP)

Certainly, Multilayer Perceptrons have a complex sounding name. However, they are considered one of the most basic neural networks, their design being:

- Input layer – layer "I"

- Hidden layer(s) – layer "H"

- Output layer – layer "O"

2) Recurrent Neural Network

The goal of a Recurrent Neural Network (RNN) is to keep the prior neuron state. This enables the neural network to maintain context and provide output depending on past states. RNNs are ideal for chatbots since preserving context during a discussion is critical to comprehending the user. A RNN model's design is shown in Figure 5.
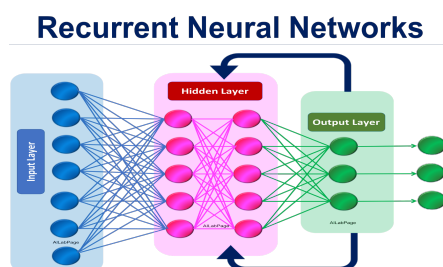

Figure 5: RNN Architecture

We have created the RNN model in following steps:

Step 1: First, we must establish a network model, which will most likely be the Sequential model: the network will be described as a series of layers, each with its own size and activation function that may be customized. The input layer will be the initial layer in these models, and it will need us to determine the size of the input we will be feeding to the network. After this more and more layers can be added and customized until we reach the final output layer (.A and Anto, 2016).

Step 2: After we've created the network's structure in this way, we must compile it, which turns the simple series of layers we've previously specified into a complicated set of matrix operations that determine how the network acts. We must specify the optimization algorithm that will be used to train the network, as well as the loss function that will be minimized, in this section.

Step 3: Once this is done, we can train or fit the network.

Step 4: The network is trained. Now we can use it to make predictions on new data.

Keras is used to create the RNN model. Keras is a high-level open source library for creating neural network models. It was created by François Chollet, a Google Deep Learning researcher. Its fundamental idea is to make the process of creating a neural network, training it, and then utilising it to generate predictions as simple as possible for anyone with a basic understanding of programming, while still allowing developers to fully customize the parameters of the ANN (Weng, 2019).

3.5 Translation of user input data

Once the model is well fitted on the data the next step in creating a chatbot is the creation of a function for translating the user input sentences to the system understandable format. Before we can begin processing intents, we need a way to produce a bag-of-words from user input. The function will process the sentences and converts it into bag of words array. A function has been created that comprises of the major data preprocessing steps and the bag of words array creation steps.

3.6 Response generation

Creating effective chatbots is a difficult task. Creating high-quality natural language replies for chatbots, in particular, is a difficult and time-consuming process that frequently relies on high-quality training data and extensive subject knowledge. As a result, it's critical to include specialists with the necessary subject expertise in the chatbot answer creation process. However, existing tool support for including domain experts in the

response creation process is limited, typically limiting users to exchanging disconnected prototypes and spreadsheets. We describe a method in this research that allows chatbot developers to efficiently involve domain experts in the chatbot answer creation process.

The easiest technique, however, still necessitates the creation of some templates in a specific language. It decreases the amount of text input while increasing the number of examples available to train the model. The creation of responses is a crucial stage in the development of a chatbot. I have created response processor using two functions they are classify and response. The intents are classified using the classify function and appropriate response are generated using the response function. Response processor consist of two main functions: Classify for intent classification and response to give appropriate response. I worked on these functions in the view of creating proper responses. Once the chatbot understands the user's message, the next step is to generate a response. One way is to generate a simple static response. Another way is to get a template based on intent and put in some variables. The steps that are involved in creating a response processor is as follows:

- Generate probabilities from the model

- Filter out predictions below a threshold

- Sort by strength of probability

- Return tuple of intent and probability

- If we have a classification then find the matching intent tag

- Loop as long as there are matches to process

- Find a tag matching the first result

- Check if this intent is contextual and applies to this user's conversation

- A random response from the intent

Response() classifies each sentence it receives. Our classifier is fast expanding and uses model.predict(). The model's probabilities are compared to our purpose specifications to provide a list of possible replies. If one or more categories exceed a certain threshold, we check to determine if a tag fits a purpose and proceed accordingly.

We'll consider our categorization list as a stack, removing items from it when we discover a good match, or until the stack is empty.

The performance of the response processor is checked using some test data and it gave an excellent performance on predicting the outputs.

4 Result

In this project I have tried implementing Multi-Layer Perceptron model and Recurrent Neural Network models on the dataset. The accuracy given by both the models are comparable, but RNN achieved an accuracy score of 97.83% whereas MLP resulted in an accuracy of 96.97%. The main benefit of RNN over feed forward neural networks is that RNN can represent a collection of records (i.e. time collection), allowing each pattern to be considered to be reliant on the preceding one. In this example, the algorithm is trained using a combination of the knowledge base and behavioral intent scenarios.

The performance of both the models are then evaluated by fitting them for predictions in the response processor. Response processor was designed in such a way that the intent classification is done using the trained models and it produces tuples with class labels and corresponding accuracy of predictions made. The below Figure 6 and Figure 7 shows the performance evaluation of the models in the response processor.

Figure 6: Predictions by RNN model.

5 Conclusion

Using natural language technology, this chatbot can help underprivileged areas by answering questions about agriculture, horticulture, and animal husbandry. Through a messaging app, the farmer

Figure 7: Predictions by MLP model

will be able to get agricultural information as well as localized information such as current market prices for specific commodities in his/her district and disease management. A farmer can send a direct message to our intelligent answering system and receive a response. Our approach would allow farmers to ask as many questions as they want, at any time, allowing current farming technologies to reach a larger number of farmers faster. Agriculture chatbots play a critical role in the agriculture industry, assisting all farmers and others interested in agricultural operations by analyzing queries and providing relevant information. This Question-Answer system is capable of answering most inquiries without the need for human interaction and with excellent accuracy. This would result in greater human resource use and the avoidance of needless expenditures associated with the establishment of additional contact centres.

Above all, I believe the method aids in the analysis of farmers' mindsets as well as the structure of India's agricultural sector. While the technology provides a safe communication route for farmers, it also aids policymakers in comprehending their wants and concerns. The data analysis also reveals which sectors or seasons demand special attention from farmers. As a result, the decision support system makes effective use of all available resources to address the problem of lack of awareness and knowledge in India's agricultural industry.

References

Raji .A and Babu Anto. 2013. Morphological synthesizer a linguistic tool for malayalam verbs. *International Journal of Computational Linguistics and Natural Language Processing*, 2.

Raji .A and Babu Anto. 2016. Knowledge base intelligent query processing. *International Journal of Latest Trends in Engineering and Technology Special Issue SACAIM*, 2:7–12.

Jeremy Beaudry, Alyssa Consigli, Colleen Clark, and Keith J Robinson. 2019. Getting ready for adult healthcare: designing a chatbot to coach adolescents with special health needs through the transitions of care. *Journal of pediatric nursing*, 49:85–91.

Vyas Ajay Bhagwat. 2018. Deep learning for chatbots.

Bob Heller, Mike Proctor, Dean Mah, Lisa Jewell, and Bill Cheung. 2005. Freudbot: An investigation of chatbot technology in distance education. In *EdMedia+ Innovate Learning*, pages 3913–3918. Association for the Advancement of Computing in Education (AACE).

Anran Jiao. 2020. An intelligent chatbot system based on entity extraction using rasa nlu and neural network. In *Journal of Physics: Conference Series*, volume 1487, page 012014. IOP Publishing.

K. K. 2020. Agrochatbot: Understanding the scope of agriculture. *International Journal of Advanced Research in Science Technology (IJARST)*, 07:6.

Rhio Sutoyo, Andry Chowanda, Agnes Kurniati, and Rini Wongso. 2019. Designing an emotionally realistic chatbot framework to enhance its believability with aiml and information states. *Procedia Computer Science*, 157:621–628.

J Weng. 2019. Nlp text preprocessing: A practical guide and template. *Towards Data Science*, 26.