

Received 15 October 2023, accepted 29 October 2023, date of publication 5 December 2023, date of current version 3 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3339253

RESEARCH ARTICLE

AgriResponse: A Real-Time Agricultural Query-Response Generation System for Assisting Nationwide Farmers

SAMARTH GODARA¹, JATIN BEDI², RAJENDER PARSAD¹, DEEPAK SINGH¹,
RAM SWAROOP BANA³, AND SUDEEP MARWAHA¹

¹ICAR-Indian Agricultural Statistics Research Institute, New Delhi 110012, India

²Thapar Institute of Engineering and Technology, Patiala, Punjab 147004, India

³ICAR-Indian Agricultural Research Institute, New Delhi 110012, India

Corresponding author: Jatin Bedi (jatin.bedi@thapar.edu)

ABSTRACT Advancements in information sciences can play a vital role in strengthening the nation's sustainable agriculture goals. In this direction, we propose a framework for a text-based query-response generation system to cope with the demand for timely help to the nationwide Indian farmers. One of the major challenges in designing such systems is constructing a knowledge base that can answer plant-protection-related questions from a diverse population of farmers. To tackle this problem, the past eight years' call-log records from the countrywide farmers' helpline network are collected and processed to construct the required knowledge base. Additionally, three response-retrieval models with approximate matching and spatial-based searching functionality are developed to administer the user input questions and extract relevant answers from the base. To validate the performance of the proposed framework, a diversified question bank consisting of 755 queries covering 151 crops in India is compiled. Three metrics (Accuracy Percentage, Crop-weighted Performance Score, and Average Response-retrieval time) are considered for the models' assessment. Experimental results show that AgriResponse is a practical framework in real-world applications, with the different retrieval models useful for different scenarios.

INDEX TERMS Artificial intelligence in agriculture, farmers' problems, helpline center knowledge base, query response generation, question answering system.

I. INTRODUCTION

Agriculture contributes most of the food and fabrics all over the globe, with a significant role in the economic development of the nations. However, due to the increasing world population, technological developments in the agriculture sector are striving to cope with the global food demand [1]. These situations call for the latest high-end technology-equipped systems to help farmers get the most out of the available resources. In this scenario, the explosion of Information and Communications Technology (ICT)-infrastructure is essential in knowledge transfer to worldwide farmers. More and more farmers are getting introduced to mobile phones daily, leading to increased

demand for helplines/support centres for their agriculture-related problems.

Furthermore, due to the large population of Indian farmers (≈ 150 million), the government has always been interested in providing help through any means possible. One such step in this direction is introducing the Kisan Call Center (KCC) program, which delivers extension assistance to the farming community through telephonic calls [2]. In 2004, the Indian government started a program to provide telephonic help to the countrywide farmers in their local language on the toll-free number "1800-180-1551". Moreover, many studies have shown the program's positive outcomes in farmers' livelihood and economic conditions over the years ([3], [4], [5]).

In the KCC, whenever a farmer calls and asks a query, the operator on the other side attempts to answer the problems


The associate editor coordinating the review of this manuscript and approving it for publication was Shadi Alawneh .



FIGURE 1. Existing workflow of the kisan call centers.

of the farmers immediately. If the KCC operator is unable to address the farmer's query instantly, the call gets forwarded to the identified agricultural specialist (Figure 1) [6]. However, as the specialists are not always available, this sometimes leads to a delayed response [7]. Keeping this scenario in mind, in the present work, we propose a text-based query-response generation model that can mimic the role of the KCC operator/agriculture specialist. The proposed model can be used to revert plant-protection-based queries from all over the country.

A. MOTIVATION AND RESEARCH GAP

This study aims to address the growing global demand for food and agricultural products due to the increasing world population. With agriculture's pivotal role in economic development, the study recognizes the need for advanced technology-equipped systems to support farmers in optimizing resource utilization. The proliferation of ICT infrastructure, including mobile phones, highlights the potential for knowledge transfer to farmers, necessitating the development of helplines and support centres. Specifically, in the context of India's large farming population, the study aims to enhance the KCC program by introducing a text-based query-response model, AgriResponse, to provide prompt and reliable assistance to farmers, contributing to their livelihoods and economic conditions. The study acknowledges the challenges of creating a comprehensive knowledge base and developing efficient response-retrieval models to handle plant-protection-related queries nationwide. It emphasizes the use of publicly available data and promotes reproducibility by making the knowledge base and models publicly accessible. Overall, the study seeks to leverage technology to enhance agricultural support and knowledge sharing, benefiting farmers and the agricultural sector's development.

The research gaps identified in this study revolve around existing question-answering (QA) models based on KCC data. While previous studies and QA models (discussed in the following section) have used KCC datasets for insights, several key flaws have been observed. Existing models address a wide range of query types, including irrelevant ones, and don't filter queries effectively. They lack mechanisms for handling spelling mistakes in the knowledge base and user queries. Furthermore, they provide only single answers,

failing to account for multiple potential solutions and don't filter out odd-lengthed or erroneous responses. These models are often tested on a district level, limiting their national-level efficiency assessment and lacking reliable performance metrics. In contrast, the proposed model focuses more on identifying correct candidate answers and accounts for multilingual data, acknowledging the challenges introduced by mixed languages and translation errors in the KCC database. The identified research gaps emphasize the need for more efficient, context-aware, and language-agnostic QA models for KCC datasets.

B. PROPOSED SOLUTION

In this study, we aim to develop AgriResponse, a system that farmers can use to ask any plant-protection-related query in textual format and get corresponding answers. The helpline operators can also use the model to gain a second opinion or explore solutions when the experts are unavailable (Figure 2). The challenges faced while designing such a system include creating a vast knowledge base to answer nationwide queries. Secondly, developing a response-retrieval model that can process mismatched words, optimizing the search time while maintaining the model's accuracy. Our solution to the first challenge is to create the knowledge base using the query-calls records stored in the KCC data servers (and on *Open Government Data Platform India* [8]) over the past eight years. The information regarding the calls is publically available by the KCCs every month. These records include information regarding the farmer's location, the farmer's query, and the answer given to the farmer, and many more.

In the present work, first, we collect the call-log files through a custom web-server crawler (over 26 million call records downloaded). Later, we perform several data-mining procedures to transform the call logs dataset into a knowledge base. Moreover, to develop an effective response-retrieval system from the knowledge base, we designed three response-retrieval models (RRMs), i.e., RRM1, RRM2, and RRM3. The developed RRMs are designed to perform approximate matches with innovative searching mechanisms.

To evaluate the developed system, we compiled a diverse question bank comprising 755 queries covering 151 crops grown throughout the country. Later, three metrics are used to evaluate the models' accuracy and response-retrieval time. Furthermore, for the reproducibility of the work,

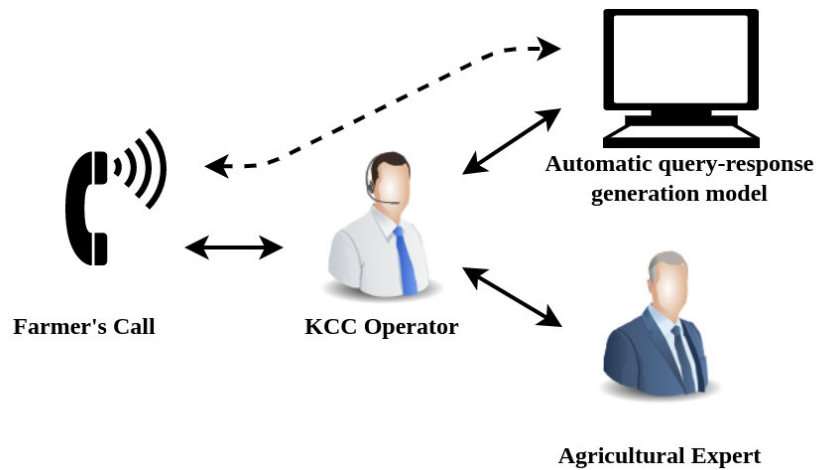


FIGURE 2. Workflow of the kisan call centers integrated with the proposed model.

we have made the developed knowledge base, the response-retrieval models' code, along with the question bank and the simulation results publicly available under GPL v3.0 through our GitHub page [9] and on Code Ocean platform [10]. Following are some of the significant research contributions of the presented study:

- **Development of AgriResponse System:** The study introduces the AgriResponse system, a novel text-based query-response generation model designed to address plant-protection-related queries from farmers nationwide.
- **Error-Tolerant Mechanisms:** It incorporates mechanisms to handle spelling mistakes in the knowledge base and user queries, enhancing the system's robustness in real-world scenarios.
- **Multiple Answer Support:** AgriResponse can provide multiple answers for a single question when necessary, recognizing that multiple solutions or treatments may be available for a given issue in different regions.
- **National-Level Efficiency:** Unlike existing models that test on a district level, AgriResponse aims to ensure its efficiency on a national level, providing more comprehensive support to a larger population of farmers.
- **Publicly Accessible Resources:** The research work promotes reproducibility by making the developed knowledge base, response-retrieval models' code, and question bank publicly available, enhancing transparency and encouraging further research in the field.
- **Incorporation of Multilingual Data:** Recognizing the multilingual nature of the KCC database, the study acknowledges the challenges of mixed languages and translation errors and tailors the model's approach to accommodate these complexities.
- **Addressing the Lack of Reliable Metrics:** By acknowledging the absence of reliable performance metrics in previous studies, AgriResponse contributes to the evaluation of QA models in this context.

The remainder of this paper is organized as follows: Section II gives the literature review regarding the works related to the present study. Section III elaborates on the methodology used to develop the knowledge base and the response-retrieval models. The experiments and results are presented in section IV, and a brief discussion of the results is given in section V. Later, section VI delivers a conclusion of the overall study.

II. RELATED WORK

The general workflow of a Question-Answering (QA) system can be divided into the following three stages [11]:

- 1) Question analysis - extracting features of the input questions and decoding what the user is asking for.
- 2) Document analysis - identifying the suitable answers to the input question.
- 3) Answer analysis - extracting and ranking the answers from the available knowledge base/corpus.

Based on the methodology used in the Question analysis stage, the QA systems can be classified into one of the three categories, i.e., linguistic approach, statistical approach, and pattern-matching approach [11]. The remainder of this section briefly describes the significant works done based on these approaches.

A. LINGUISTIC APPROACH-BASED QA MODELS

In the linguistic approach-based QA models, first, techniques such as tokenization, POS (Part-of-speech) tagging, and parsing are used to process the user's questions. Later, the corresponding solutions are reverted according to the "meaning" of the asked question. The limitation of such systems is that the information stored in the structured database could only counter questions asked within the limited speciality. Researchers started developing systems based on this approach in the 1960s [12]. The early designs used natural language processing techniques to produce canonical forms, which in the later steps were used to

form a standard query for database search. Later in the 1970s, a few other similar models were developed based on dialogue systems ([13], [14]), using structured databases as the knowledge source. Moreover, in 1999, Clark et al. [15] introduced a method for integrating online content with knowledge-based question-answering ability. This mixed strategy allowed users to locate common and random questions at the time of system development. Furthermore, many models were developed at this time, which used online text as their knowledge resource ([16], [17], [18]). These models used their heuristic functions to save information from the internet to the local knowledge database records.

In 2011, Moreda et al. [19] developed a QA system based on semantic information, semantic roles, and WordNet. They concluded that the more semantic knowledge the system uses, the better the accuracy the model achieves. Hristovski et al. [20] developed a biomedical QA system based on semantic relations extracted from biomedical literature in 2015. In the same year, Huang et al. [21] developed a QA system based on Wikipedia data extraction. In this work, the researchers used Natural Language Processing (NLP) techniques, including NER (Named Entity Recognition), POS, and dependency parsing. Moreover, similar techniques were used by Xie et al. [22], including POS tagging, syntactic analysis, and semantic relation query keywords to develop a QA system based on ontology.

B. STATISTICAL APPROACH-BASED QA MODELS

Models based on statistical methods analyze the users' questions to make predictions about the expected answers. These methods analyze the documents based on several similarity features to define the closeness of applicant answers to the input question. The models are generally trained on a corpus (manually or through machine learning algorithms) annotated with the specific categories. Some common techniques used in these models include SVM, Naive-Bayes, K-nearest neighbour, etc. [23]. In 2000, IBM's researchers developed a statistical QA system which used a maximum-entropy model for text classification based on the bag of words features [24]. Later in the 2000s, many other text-classification-based models were developed using similar techniques ([25], [26], [27]). Furthermore, Soricut et al. [28] developed a statistical chunker-based model to chunk the input questions into phrases asked to the search engine. The developed system could also answer difficult questions, using N-gram co-occurrence statistics to extract the most suitable answer.

In 2005, Higashinaka et al. [29] developed corpus-based QA for why-questions using ML-based techniques to train a ranker of answer candidates on the basis of features. Later, in 2009, researchers introduced a QA system using an improved Bayesian method based on ontology to classify the questions [30]. Toba et al. [31] proposed the approach of using a hierarchy of classifiers to discover high-quality answers in community QA archives

in 2013. Moreover, in 2016, Medved et al. [32] developed an Automatic QA system for Czech using a combination of TF-IDF and tree distance between the question and candidate answers. Similarly, Lima et al. [33] developed a multi-level tag recommendation integrated with an external knowledge bases QA system, which internally categorises the tags into different semantic levels based on their usage frequencies. Furthermore, Ha et al. [34] developed a QA system for medical MCQs, combining a neural approach with an information retrieval approach in 2019. In 2020, Oniani et al. [35] developed and evaluated various language models for Covid-19-related QA systems using the GPT-2 language model and applied transfer learning to retrain it on the Covid-19 open research dataset. Furthermore, in 2021, Aithal et al. [36] developed a QA pairs generation system and introduced a question similarity mechanism that imitates human reasoning to identify whether the questions posed are answerable.

C. PATTERN-MATCHING APPROACH-BASED QA MODELS

In the pattern-matching approach-based models, the responses to the input questions are recognized based on the similarity between their structural patterns, including specific semantics. In these systems, surface pattern-based methods are used to obtain factual responses, as those answers are restricted to fewer sentences. In 2002, Ravichandran et al. [37] introduced a bootstrapping-based automatic learning method to construct a large set of patterns. In the same year, Zhang et al. [38] introduced the concept of surface patterns combined with the "support" and "confidence" from data mining. Later, in 2003, Greenwood et al. [39] integrated named entity tagger with the surface patterns to generalize the patterns produced from the loose text. Furthermore, in 2007, Cui et al. [40] utilized the bigram model and Profile Hidden Markov Model-based soft pattern matching to identify the responses. Saxena et al. [41], in the same year, used a similar approach for complicated questions, including date of birth, location, and acronym expansion-related questions.

D. COMPARISON WITH THE EXISTING FARMERS' HELPLINE-BASED QA SYSTEMS

Although several studies have been done on the KCC dataset for extracting useful insights ([42], [43], [44], [45], [46]), recently, two research teams have developed QA models similar to this present study ([7], [47]). Data for these QA systems is collected from the KCC data servers, which is later processed through NLP. However, there seem to be the following major flaws in these existing models:

- There are several types of call-log records present in the KCC data servers, including queries regarding weather ($\approx 40\%$), plant-protection ($\approx 22\%$), market information ($\approx 3\%$), seed ($\approx 1\%$), etc. Moreover, the answers corresponding to many previous queries asked by the farmers do not have any value in the future.

Whereas the existing systems handle all types of queries, including weather and crop-price-related queries, which such systems shouldn't address.

- Since the records are manually fed into the KCC system, it was noted that there exist some spelling mistakes in the knowledge base. Therefore, a system with an exact matching mechanism may not be a good design choice for this knowledge base. Moreover, no such mechanism is used for handling the users' spelling mistakes in the existing models; neither are the misspelt words present in the dataset dealt with.
- There can be multiple answers present corresponding to a single question in the database. The reason behind this is that the availability of the control treatment differs from place to place. Secondly, there can be multiple treatments available in the market for the same disease/pest. Moreover, it was found that all the existing models are designed to revert to only a single answer corresponding to the input question.
- From the observations, it was noted that different-sized answers are present in the database. Moreover, the large-sized solutions (more than 100 characters) and small-sized solutions (less than eight characters) present in the database generally contain errors (unwanted extra symbols, long continuous series of alphabets). Therefore, it is better to filter out odd-lengthed answers from the database. Whereas no length-based filters are used in the existing models to filter out the erroneous answers.
- The testing of the developed models is done on the district level in each of the studies, which does not ensure the model's efficiency on the national level.
- No reliable metric is used to assess the performance of the models in the studies.

Moreover, compared with the existing QA models, the proposed model invests less in processing the input question and more towards finding the correct candidate answers. This is because it is found that there are several entries (questions) in the KCC database in other languages, i.e. questions which are in Hindi (or other local languages), typed in the English language by the KCC operators. Moreover, there is a possibility of induction of errors in text translations from query-call voice. Because of the contamination of the corpus (mixed languages and translation errors), designing a conventional NLP system on such a knowledge base is not a good fit. Consequently, AgriResponse does not expect the input query in natural language; it basically decomposes the single-sentence query into the following three questions:

- What is the location of the farm?
- Which crop is being asked about?
- Which disease/pest is to be controlled?

Each of the above questions has one-word (or phrase) answers, and since these three parameters are needed to solve the problem, it is better to ask the parameters separately. This design helps optimise the accuracy of

the later processing steps, as the error is minimized while processing the input questions. Another novelty of the presented work is the location-based smart searching mechanism of the response retrieval models to optimize the average response retrieval time. Furthermore, the proposed model uses Lavenstien distance (LD) in the later stages to search for the closest questions in the database.

III. METHODOLOGY

The present research work can be divided into two parts, i.e. knowledge base construction and development of the response-retrieval models. The knowledge base developed in the presented work comprises the farmers' helpline call logs. Moreover, these call logs consist of several attributes, including the questions the farmer asks, the answers delivered by the helpline operator, and many more (Table 1). Furthermore, we developed three Response Retrieval Models (RRMs) to retrieve suitable answers corresponding to the questions asked. The developed retrieval models in this study take three separate parameters as input, i.e. state name, crop name and disease/pest name, to search the candidate answers.

The first retrieval model, RRM1, uses only the input's crop name and disease/pest name parameter. Moreover, the model's output consists of the top 5 answers in the knowledge base corresponding to the input query. The second model, RRM2, is built using RRM1; in addition to the two parameters, it also considers the location of the query (state name). RRM2 selects the dataset based on the input state name and passes the other parameters of the query along with the selected dataset to the RRM1. A similar input pattern is followed by the RRM3 (input includes the state name, crop name, and disease/pest name entered by the user); moreover, the model smartly selects the statewide records to be searched in the knowledge base. RRM3 searches for the answers in the knowledge base starting from the closest state. The model uses a distance matrix for all the states so that whenever a new query is asked, the model generates a sorted list of states in ascending order to the distance of the state of the query. Later, the selected state name, the crop name and the disease/pest name are passed to the RRM2, and the answers are retrieved using RRM1.

A. KNOWLEDGE BASE CONSTRUCTION

The procedure followed for the construction of the knowledge base can be divided into two basic steps, i.e., data collection and data preprocessing (Figure 3).

- 1) Data collection: In order to fetch the call-log files from the KCC Helpline's data servers, a dedicated web crawler is designed. The data servers maintain a single file corresponding to a single district of India for a single month. As the whole dataset is available in 55,844 .json formatted files, downloading the dataset manually is not a feasible task. The custom-designed web crawler iteratively explores all the districts and

TABLE 1. Attributes’ description of the downloaded data.

Attribute	Sample Values	Description
QueryText	asked about reddening of leaves, control of tsetse fly, asked about sucking pest, leaf webber	Query made by the farmer
KccAns	spray micromax, spraying of carbaryle 3 grms/lit water suggest to spray chloripyriphos, carbaril 2g/l	Query response by the KCC operator
QueryType	Plant Protection, Water Management, Field Preparation, Agriculture Mechanization	Query type
Category	Pulses, Cereals	Crop category of the asked query
Crop	Wheat, Maize	Crop being asked about
Sector	AGRICULTURE, HORTICULTURE	Farming sector of the query
Season	RABI, JAYAD, KHARIF	Cropping season of the year
CreatedOn	2019-11-04 08:32:00.000	Year, month, date and time of the query (temporal information)
StateName	UTTAR PRADESH, WEST BENGAL, ANDHRA PRADESH, BIHAR, PUNJAB	State of the farmer (spatial information)
DistrictName	Barnala, Bathinda, Faridkot, Fatehgarh Sahib, Fazilka, Ferozepur	District of the farmer (spatial information)
BlockName	Jandiala Guru, Majitha, Rayya, Tarsikka, Verka	Block of the farmer (spatial information)

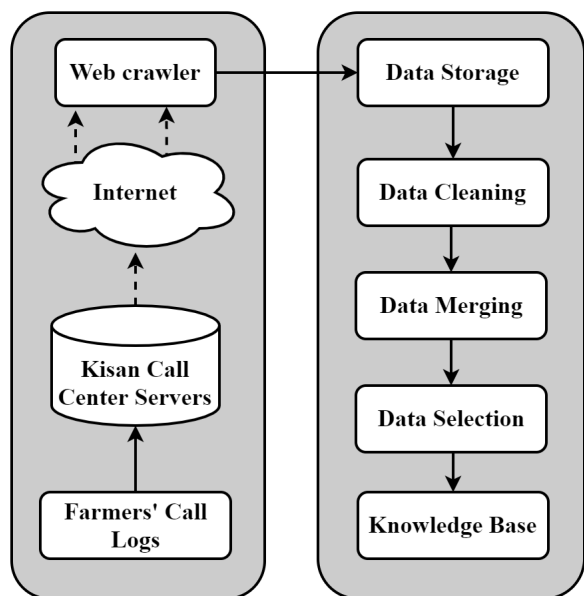


FIGURE 3. Methodology used for construction of the knowledge base.

requests data from the servers regarding all possible locations in India. Overall, 26,874,198 call-log records between the period of March 2013 till February 2021 are downloaded from the data servers and are forwarded to the data preprocessing module. The details regarding the attributes of the call-log records, along with their sample values, is given in table 1.

2) Data preprocessing: The data preprocessing step can be further divided into the following three sub-steps:

- a) Data cleaning: it is used to remove any erroneous data/characters from the downloaded call-log files. Such characters get introduced to the dataset due to human input error or improper storage or transfer processing. In our study, we eliminated all the characters from the tuples except the alphabets (lower case and upper case), the numerical characters and some other selected characters, including commas, colons, etc. Examples of strings before and after the cleaning step are given in table 2.
- b) Data Merging: As the call log records are scattered in numerous files, it is required to merge the complete data into a single file. This action makes it easier to handle data in the next steps. For this step, a programming loop is designed to go through all the downloaded (55,844) files, read the tuples and append each tuple in a single destination file. The output of the step is a single file, with the records from all the multiple files, including all the attributes shown in table 1.
- c) Data Selection: The dataset obtained until the previous step contains several irrelevant attributes. Since we aim to develop a query-response system specifically for the plant-protection-related questions, we first filter out all the data records which do not belong to this category. The process is to keep only the records with the exact match of the string “Plant Protection” in the “QueryType” attribute of the dataset (Table 1). Moreover, since only four attributes are used in the response retrieval models (CropName, StateName, QueryTest, and KCCAns),

TABLE 2. Input and output samples of strings before and after data cleaning process.

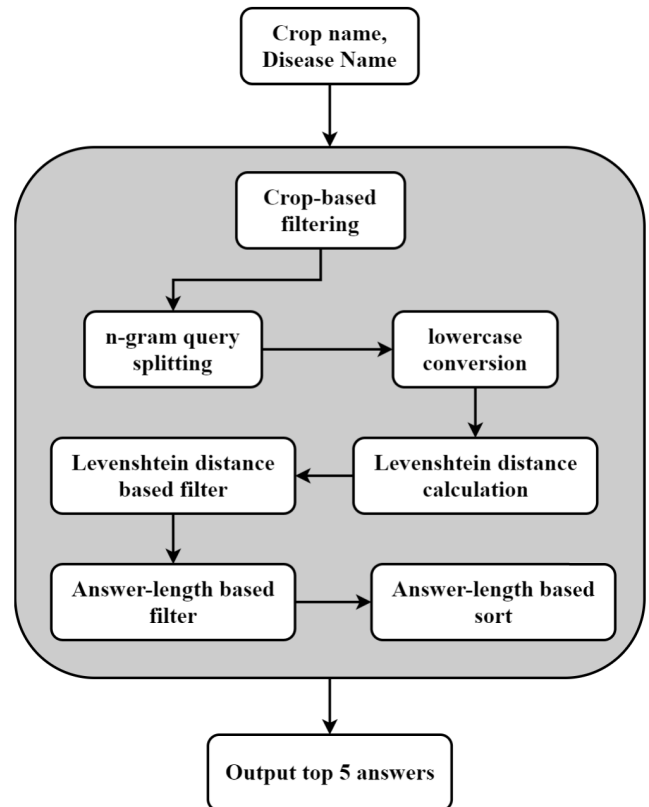
S. No.	Input String	String after cleaning
1.	asked about reddening of leaves.	asked about reddening of leaves
2.	suggest to spray -chloripyriphos;	suggest to spray chloripyriphos

rest of the attributes are removed from the “Plant Protection” dataset. At last, we end up with 5,921,883 records in our knowledge base, ready to serve information to the response-retrieval models. The final knowledge base developed in the study is publically available and can be downloaded from our online repository- <https://github.com/Samarth-Godara/AgriResponse> [9].

B. RESPONSE-RETRIEVAL MODELS CONSTRUCTION

The remainder of the section elaborates on the three RRM developed in the present study. Each of the three models takes three components as input, i.e. state name, crop name, and disease/pest name. Moreover, the RRMS are designed in such a way that the input state name and crop name must be an exact match with any of the records present in the knowledge base. This is done because both these parameters have separate attributes to match the information from (“StateName” and “Crop”, table 1). Moreover, there are limited states and crops in India; therefore, these parameters can be inputted using drop-down boxes where there is no possibility of spelling mistakes. On the other hand, the models are designed to perform approximate matching of the input disease/pest name. The structure of the models’ outputs is similar; all the models output the most suitable answers related to the input query parameters in the form of an array of five strings. The Python programs of the models developed under the study are available online [9]; following is the detailed description of the step-by-step internal working of the developed RRMs.

- 1) Response-Retrieval Model 1 (RRM1): The RRM1 is developed to only use crop names and disease/pest names in order to search for answers in the knowledge base. The workflow followed by the model (Figure 4) to extract the suitable answers is as follows:
 - a) Crop-based filtering: The first step is to filter all the answers corresponding to the crop in the input parameters. This is done by matching the input crop string with the data values from the “Crop” attribute present in the knowledge base. The step helps remove the irrelevant rows from the search domain, which consequently helps in faster computation of the subsequent steps. The dataset after the first step is processed row by row, where the question present in each record is processed separately; therefore, it is necessary to drop the baggage as much as possible.
 - b) N-gram query splitting: In this step, the “QueryText” of each tuple present in the selected dataset is split into

**FIGURE 4.** Block diagram of response-retrieval model 1.

all possible n -grams, where n is the number of words present in the disease/pest parameter of the input query. This step is performed because each n -gram of the question will be compared with the disease/pest word in the later steps. Moreover, if the disease/pest word has two words (for example, ‘leaf miner’), the searching algorithm must compare the input with the sets of all two consecutive words from the question. The algorithm 1 shows the procedure used for n -gram extraction from a given “question” string. Examples of the input and output of the algorithm are given in table 3.

- c) Lowercase conversion: In this step, all the characters present in the input disease/pest name and the “QueryText” strings of the selected dataset are converted into lowercase. This process is simple, yet it significantly helps with the consistency of the expected output, as it eliminates the scenarios of mismatches due to different casings of the characters.

TABLE 3. Input and output examples of the n -gram extraction algorithm.

S.No.	Input Question String	n	Output
1.	The farmer enquired about the treatment of leaf miner in kinnow	1	“The”, “farmer”, “enquired”, “about”, “the”, “treatment”, “of”, “leaf”, “miner”, “in”, “kinnow”
2.		2	“The farmer”, “farmer enquired”, “enquired about”, “about the”, “the treatment”, “treatment of”, “of leaf”, “leaf miner”, “miner in”, “in kinnow”,
3.		3	“The farmer enquired”, “farmer enquired about”, “enquired about the”, “about the treatment”, “the treatment of”, “treatment of leaf”, “of leaf miner”, “leaf miner in”, “miner in kinnow”,

Algorithm 1 Algorithm for n -Grams Extraction

Input : question string, disease/pest name string
Output: bag (set of all n -grams extracted from the input question string)
 $list \leftarrow tokenize(\text{question string});$
 $n \leftarrow size(tokenize(\text{input disease/pest name}));$
 $bag \leftarrow \emptyset;$
 $i \leftarrow 0;$
while $i < (size(list) - (n - 1))$ **do**
 $w \leftarrow list[i : (i + n)];$
 $bag \leftarrow bag + w;$
 $i \leftarrow i + 1;$
end

- d) Levenshtein distance (LD) calculation - In this step, the LD is calculated between the input disease/pest name and the n -grams extracted from each of the records of the selected dataset. In later steps, questions are filtered out based on the minimum LD value. Commonly called the ‘edit distance’, LD is a string metric used to measure the difference between two sequences. LD is a distance metric widely used for approximate matching tasks [48]. Informally, the LD between two phrases is the minimum amount of character edits (insertions, deletions, or replacements) needed to change one phrase into the other. This distance metric helps find the matching words from the input query. Moreover, one of the methods used to calculate the LD between two given strings is described in algorithm 2 (output sample given in table 4).
- e) Levenshtein distance-based filter- In order to filter out the records carrying the questions which do not contain any word (phrase) close to the disease/pest parameter of the input query, we discard all the records which have LD values more than 2. Table 5 gives examples of questions filtered based on the closest LD value, and figure 5 explains the mechanism of calculating LD between a knowledge base question and the user input disease/pest name with an example.

Algorithm 2 Algorithm to Calculate Levenshtein Distance Between Two Input Strings

Input : $str1$, $str2$, m , n (two input strings along with their lengths)
Output: LD (Levenshtein Distance between the two input strings)
if $m = 0$ **then**
 $LD \leftarrow n;$
else if $n = 0$ **then**
 $LD \leftarrow m;$
else if $str1[m - 1] = str2[n - 1]$ **then**
 $LD \leftarrow LDistance(str1, str2, m - 1, n - 1);$
else
 $LD \leftarrow (1 + \min(LDistance(str1, str2, m, n - 1), LDistance(str1, str2, m - 1, n), LDistance(str1, str2, m - 1, n - 1)));$
end
return LD

TABLE 4. Examples of LD values between different strings.

S. No.	Input string 1	Input string 2	LD
1.	“leaf miner”	“farmer”	6
2.		“leaf minor”	2
3.		“queiry”	9

- f) Answer-length-based filter: As discussed in previous sections, it is observed that too short and too long answers present in the knowledge base tend to contain erroneous information. Therefore, we filter out the very lengthy answers. These records contain random sequences of alphabets, which could have been introduced by some human or digital error (storage or transfer error). In order to remove such records from the dataset, we discard the answers containing more than 100 characters.
- g) Answer-length-based sort: After filtering the answers based on their lengths, the responses are sorted in descending order according to the same. This is done

TABLE 5. Examples of LD-based question filtering.

S.No.	Disease/pest parameter	Sample Question	Closest LD in sample question	Decision
1	leaf miner	farmer requesting information regarding weather	12	discard
2		problem of blast in wheat, treatment asked	7	discard
3		advice for problem of leaf minor in kinnow orcard	2	keep

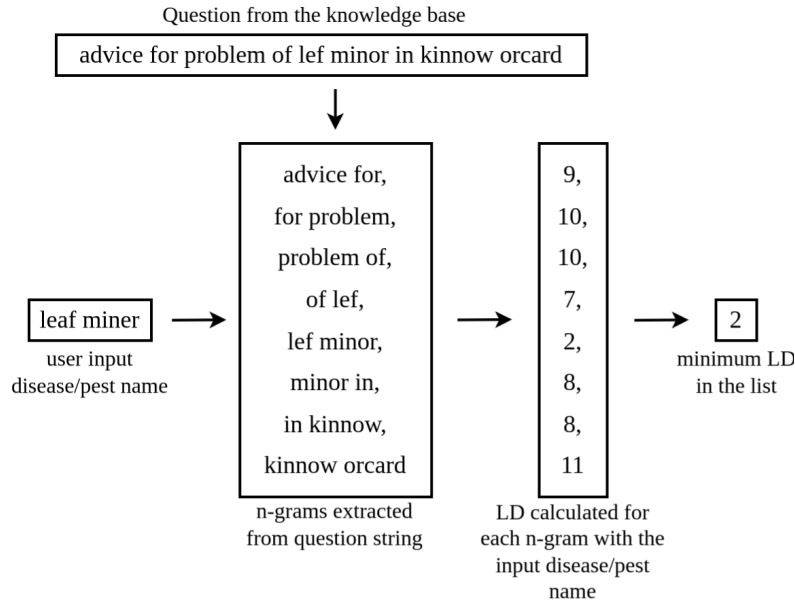


FIGURE 5. Example of LD calculation between a question from the knowledge base and the user input disease/pest name.

so that the answers of appropriate lengths come at the top for the retrieval.

- h) Model’s Output: In the last step of the RRM1, the top five answers obtained after sorting are reverted. This model does not use spatial information (state name). Consequently, it searches for the best answers in the countrywide dataset. A few examples of the input parameters and output of the model corresponding to them are given in table 6. As shown in example 1 of table 6, the last answer is erroneous. Whereas all the other four output answers are acceptable.
- 2) Response-Retrieval Model 2 (RRM2): RRM1 does not consider the spatial information present in the query parameters, whereas RRM2 uses this information to deduce the search time. In order to do so, RRM2 first filters out the records which do not belong to the input state name. A simple exact-matching module compares the input state-name-string with the data values present in the “StateName” attribute of the knowledge base. Later, the model feeds the filtered dataset to RRM1

along with the other input parameters (crop name and disease/pest name) (Figure 6). This helps in cutting down the processing time of the following steps, consequently affecting the model’s accuracy (discussed in section IV). The output of RRM2 is similar to the output of RRM1 (Table 6).

- 3) Response-Retrieval Model (RRM3): RRM3 is an extension of RRM2, where RRM3 smartly switches states if the answer is not obtained from the current state dataset. First, the geolocation of the approximate centres of all the states is obtained manually and fed to the geo-matrix construction module. Later, if no answer is found in the input state name, the RRM3 iteratively uses the geo-matrix to choose the nearby states to search for the answers in (Figure 7). After the evaluation of the models’ performances, it was found that RRM3 performs better in terms of accuracy than RRM2. A detailed stepwise working of the RRM3 model is explained below.
 - a) State-wise geolocation matrix construction: First, the approximate geolocation (latitude and longitude)

TABLE 6. Sample input and output of RRM1.

S.No.	Model's Input	Model's Ouput
1.	State Name	RECOMMENDED TO DRENCH WITH BLEACHIG POWDER 20GLITERPLANTOR CARBENDAIZM 3GLITERPLANT
	MAHARASHTRA	recommended for drenching with copper oxy chloride 25 gm lit of water
	Crop Name	recommended for soil drenching with carbendazim 2 gm lit of water
	Banana	recommended for carbendazim 2 gm lit of water
	Disease/ Pest Name head rot	50 500gm 200
2.	State Name	RECOMMENDED TO APPLY IMIDACLOPRID 1 ML IN 3 LITRES WATER TO CONTROL SUCKING PEST IN BEANS
	MAHARASHTRA	sucking pest on rajma beanspray tata mida 10ml bavistin 30gm 15 lit of water
	Crop Name	Spray Actra 5 gm 15 Lit of WaterSpray Redomil Gold 30gm 15Lit of Water
	Rajma (french bean)	Sucking Pest on beanSpray Confidor super Confidor 10ml15 Lit of Water
	Disease/ Pest Name sucking pest	to control sucking pest on rajma Spray Confidor 10ml15 Lit of Water
3.	State Name	RECOMMENDED TO SPRAY MANCOZEB (M45) 600 GRAMS 200 LITRES OF WATER ACRE 600 200
	TAMILNADU	RECOMMENDED TO SPRAY MANCOZEB (DITHANE M45 INDOFIL M45 MANJET) 500600 GRAMS 200 L WATER PER ACRE
	Crop Name	RECOMMENDED TO SPRAY MANCOZEB (M45) 600 GRAMS 200 LITRES OF WATER ACRE 600 200
	Maize (Makka)	Recommended to spray Mancozeb 75WP 400gacre200liter (2g 1liter) of water to control Rusts in maize
	Disease/ Pest Name rust	RECOMMENDED TO SPRAY MANCOZEB 75WP(DITHANE M45)3 GRAM PER 1 LITRE WATER TO CONTROL RUST IN MAIZE

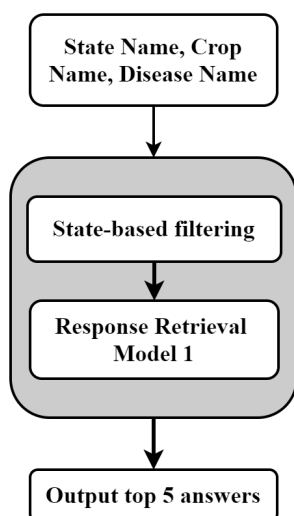


FIGURE 6. Block diagram of response-retrieval model 2.

information is gathered regarding the centre points of India's states/union territories. Later, a matrix is constructed with each row and column representing each of the Indian states/UTs (Figure 8). Each entry of the matrix contains the Euclidean distance

between the states representing the row and column of the cell.

$$d_{ij} = \sqrt{(glat_i - glat_j)^2 + (glon_i - glon_j)^2} \quad (1)$$

Here, d_{ij} represents the cell belonging to the i^{th} row and j^{th} column, $glat_i$ and $glon_i$ represent the latitude and longitude of the i^{th} state, respectively. It is to be noted that construction of the geo-matrix is done once, and the same is used for retrieving information corresponding to the subsequent input queries.

- b) Model's Input: RRM3's input and output parameters are similar to those of the other models. Moreover, RRM3 searches for answers in a smart state-wise manner. Furthermore, it can be deduced that RRM3 behaves similarly to RRM2 in the first iteration.
- c) Distance-based sort: The model generates a sorted list of states in ascending order of the distance from the input state name. It is done by sorting the values of the column corresponding to the input state name in the geo-matrix. An example of the state-wise sorting is shown in figure 9, where the state name in the input query is Gujarat (dark red coloured state, western side of India), and the states are coloured in lighter shades

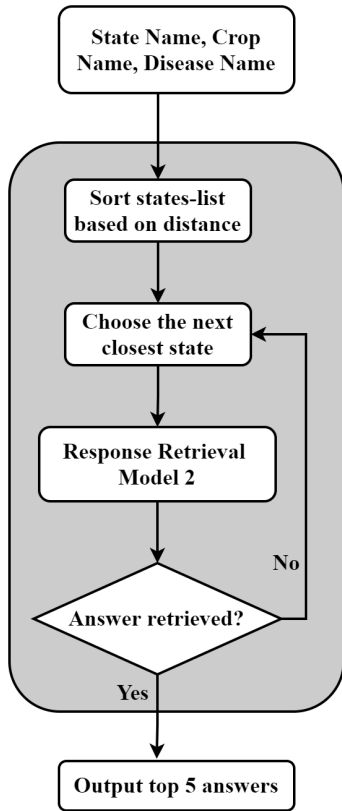


FIGURE 7. Block diagram of response-retrieval model 3.

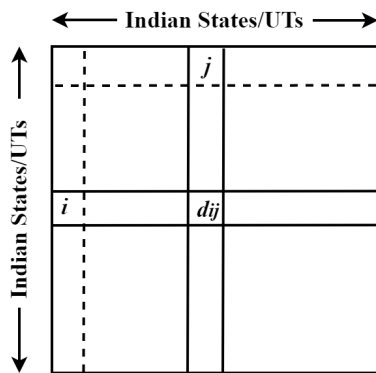


FIGURE 8. Structure of the geo-matrix.

of red as the sequence of the search algorithm moves to the next state. It can be noted that the search started from the western part of India and moved towards the eastern parts after covering northern, central and southern India.

- d) Switching states: The state-search list helps RRM3 search state-by-state in the countrywide dataset. The first state in the list is the one which has been fed to the model as the input. Later, the selected state name is fed to RRM2 along with the other parameters. RRM2 performs the search in the input state name dataset and reverts back to the available answers. If no answers are retrieved from the selected state dataset, the next

state name is fed to RRM2 from the list. Although this state-search technique seems to be complex, it gave better results in terms of response-retrieval time than RRM1 (discussed in section IV).

- e) Model’s output: Each model developed in the present work has a similar structure of output, i.e. each model reverts a list of five strings (answers) corresponding to the input state name, crop name, and disease/pest name (Table 6).

IV. EXPERIMENTS AND RESULTS

In order to evaluate the performance of the developed models, a question bank is compiled consisting of 755 questions covering 151 crops grown in different regions of India. Five queries are collected corresponding to each of the 151 crops in the question bank, with different combinations of the state name and disease/pest name.

The question bank developed for assessing the proposed framework is available online [9]. The whole process is simulated with Python 3.0 script on the Google Colab platform with dual Intel(R) Xeon(R) CPU @ 2.20GHz microprocessor, 13GB RAM and 108GB disk space. In the experiments, each model is automatically fed the queries one by one from the question bank (Figure 10). Later, the models’ responses are examined through the below-mentioned three metrics:

- 1) Accuracy Percentage (AP): AP represents the percentage of questions corresponding to which the model has outputted at least one correct answer and no incorrect answer (equation 2).

$$AP = \frac{1}{755} \sum_{j=1}^{755} (a(R_j)) \times 100$$

$$a(R_j) = \begin{cases} 1, & \text{if there is at least one correct answer} \\ & \text{in the list and no incorrect answer} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where R_j represents the response from the model corresponding to the j^{th} query of the question bank.

- 2) Crop-weighted Performance Score (CWPS): CWPS represents the AP estimate of the model in a real-life scenario. As the farmers do not ask queries regarding all the crops with the same frequency, the correct answers corresponding to a frequently asked crop must carry more weightage than the others. Considering this situation, we propose a new metric that can be used to infer the percentage of correct answers the models would revert in real-world scenarios. In this process, first, the weight of each crop is calculated by equation 5, which is the fraction of occurrences of the queries related to the target crop in the knowledge base. Later, the percentage of the weighted score is calculated as

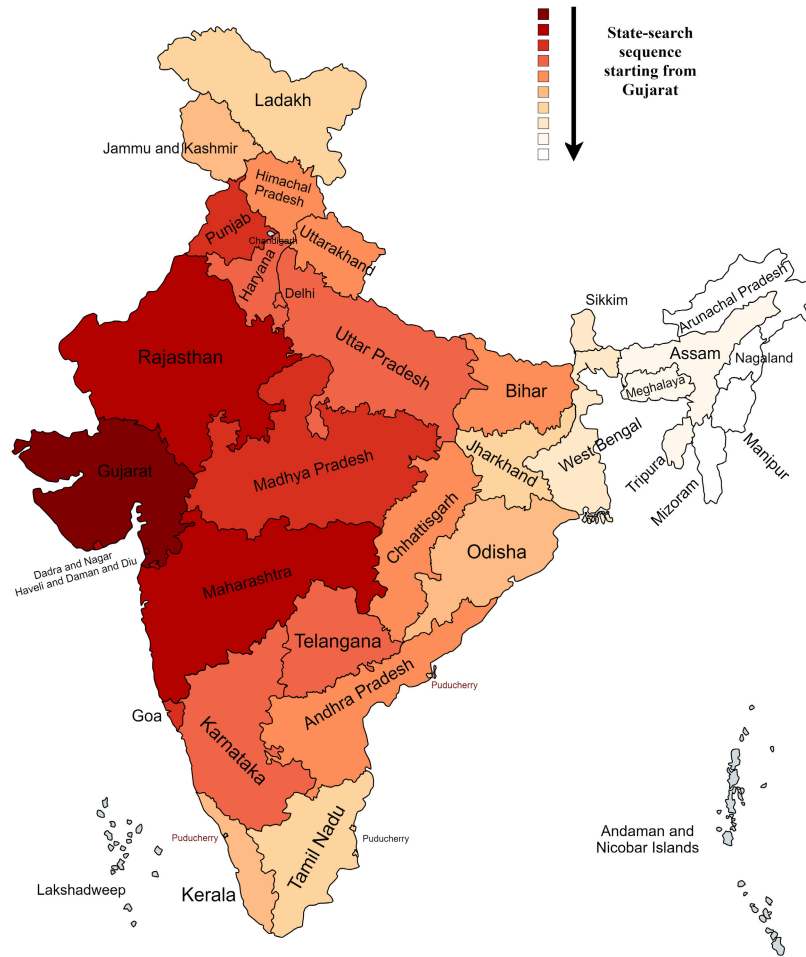


FIGURE 9. Example of the state-search sequence where *Gujarat* is inputted as the state name in the input query. The search sequence starts from the darker shade of red and moves to the states with the lighter shades of red, i.e. starting from Western India, covering Northern, Central and Southern India, and at last, Eastern India.

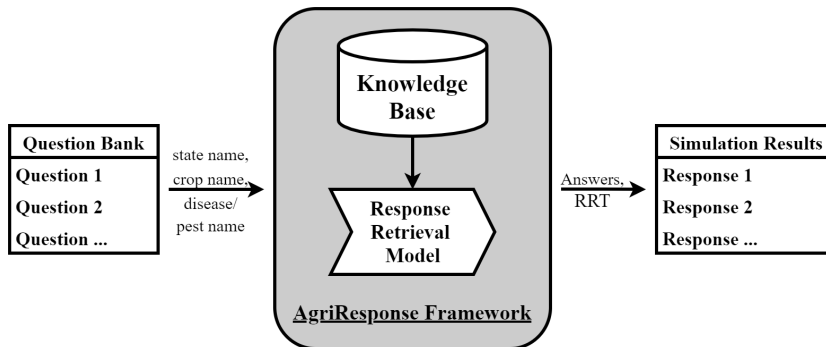


FIGURE 10. Simulation process for agriResponse framework.

shown in equation 3.

$$CWPS = \frac{1}{755} \sum_{j=1}^{755} (s_j) \times 100 \quad (3)$$

$$s_j = \frac{a(R_j) \times w_j}{5} \quad (4)$$

$$w_j = \frac{n_j}{T} \quad (5)$$

Here, n_j represents the total number of questions present in the knowledge base corresponding to j^{th} crop and T represents the total number of questions present in the knowledge base.

- 3) Average Response-Retrieval Time (ARRT): ARRT represents the average time consumed (in seconds) by the model to revert back with the list of answers (or an empty list) corresponding to the input query parameters. It is

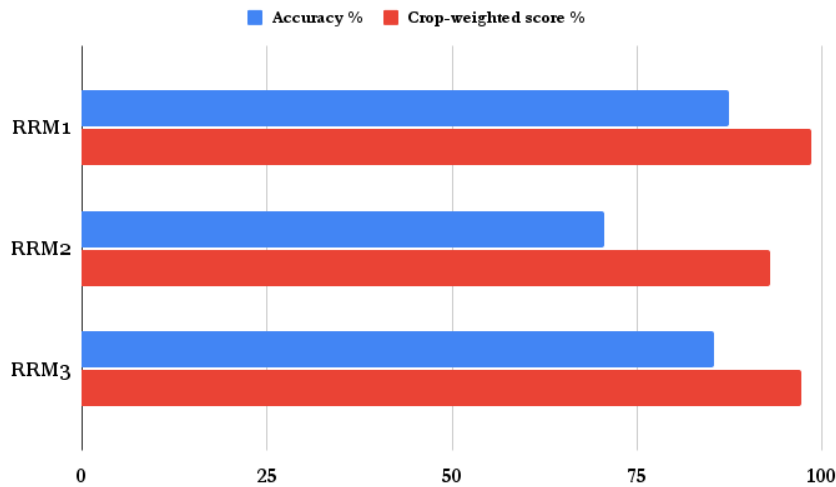


FIGURE 11. Accuracy percentage and crop-weighted performance score of the RRM models.

calculated using equation 6.

$$ARRT = \frac{1}{755} \sum_{j=1}^{755} (t_j) \quad (6)$$

Here, t_j represents time consumed (in seconds) by the model to revert back the response corresponding to the j^{th} question of the question bank.

A. ACCURACY PERCENTAGE (AP) AND CROP-WEIGHTED PERFORMANCE SCORE (CWPS) RESULTS

A comparison of the AP and CWPS of the three RRM models is given in table 7 and figure 11. It can be noted that the performance of RRM3 is very close to that of RRM1 in both the metrics (RRM1 performed $\approx 2\%$ better than RRM3). From the CWPS of RRM1 and RRM3 ($\approx 97\%$), it is inferred that these models will probably revert out only three wrong answers out of 100 asked questions in real-world scenarios. Whereas the performance of RRM2 is very low in terms of AP (70.59%) as compared to other models ($\approx 85\%$). Moreover, the CWPS of RRM2 is in comparison with the other models, although performance is still lower than RRM1 and RRM3.

TABLE 7. Accuracy percentage and crop-weighted performance score of the RRM models.

Model	Accuracy %	Crop-weighted Performance Score
RRM1	87.41%	98.5%
RRM2	70.59%	92.91%
RRM3	85.43%	97.15%

The reason behind the low performance of RRM2 is the fact that RRM2 searches for the answers only in a single-state dataset. Although RRM3 also performs a single-state search, it covers the whole country state-by-state iteratively. Therefore, RRM3's performance is close to RRM1, which directly

searches in the nationwide knowledge base. The reason behind the far better CWPS of the models compared to AP is that the models are not able to answer the queries which are less frequently asked in the past years. Therefore, there are comparatively fewer answers present in the knowledge base regarding those crops.

B. AVERAGE RESPONSE RETRIEVAL TIME (ARRT) RESULTS

Table 8 and figure 12 elaborate on the models' performances in terms of ARRT. It is observed that RRM2 performs the best among all the models, with an ARRT of 2.29 seconds. This is a predictable observation because RRM2 has the smallest search space as it searches only into a single state-wise knowledge base section. Moreover, RRM3 showed an 82.53% increment in ARRT, whereas RRM1 showed a 406.55% increase in ARRT compared to RRM2.

V. DISCUSSION

Timely help is the requirement of the hour for worldwide farmers. This generates the need for an automated answering system which can assist the diverse population of farmers. The existing helpline systems are human-driven, leading to delays in help to the farmers. Considering the current scenario, the proposed framework, AgriResponse, can use a knowledge-based query-response retrieval system to help the KCC operators in a semi-automatic way. For the objective, initially, a knowledge base is built using call-log records from the country-wide farmers' helpline centres. Three response-retrieval models are developed to retrieve information from the compiled knowledge base (RRM1, RRM2, and RRM3). The results show that the models are useful in various scenarios according to the requirement of the system, i.e. accuracy or RRT.

In order to have a thorough understanding of the proposed system's behaviour, we investigated more in the direction of the frequency distribution of the number of queries present

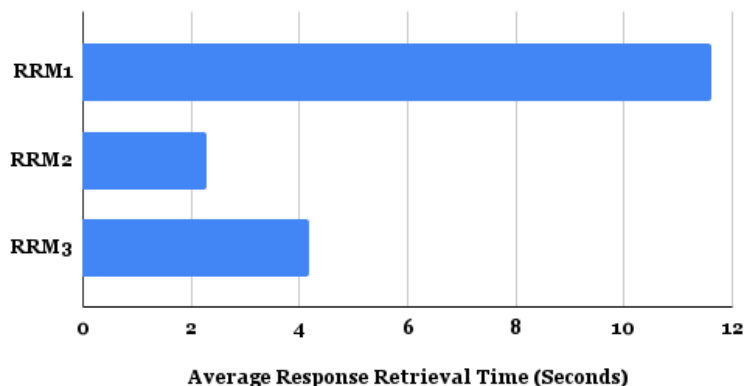


FIGURE 12. Average response-retrieval time performances of the models.

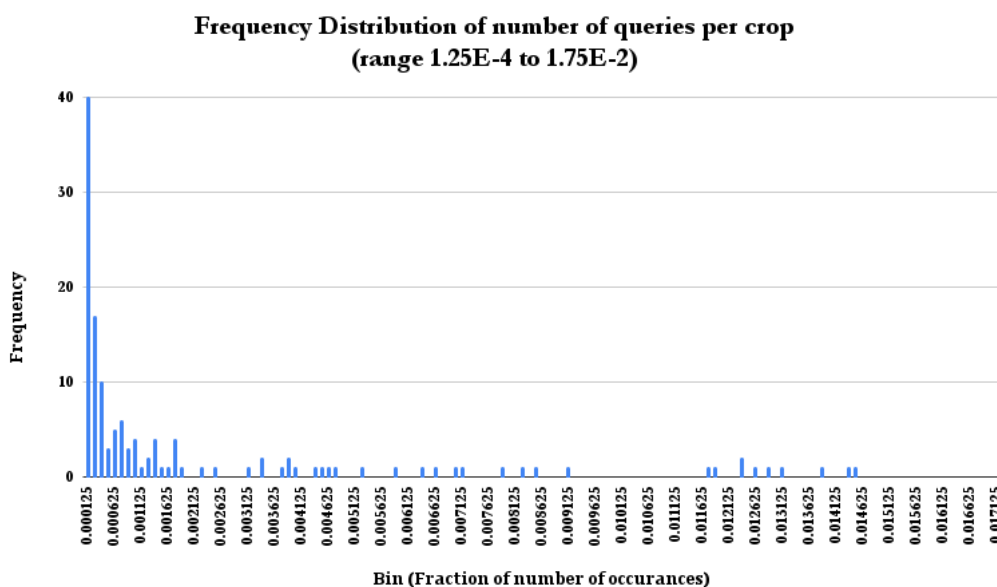


FIGURE 13. Frequency distribution of the number of queries per crop in the knowledge base.

TABLE 8. Average response retrieval time performances of the RRM’s.

Model	Average Response Retrieval Time
RRM1	11.60 seconds
RRM2	2.29 seconds
RRM3	4.18 seconds

in the knowledge base corresponding to each crop. Figure 13 shows the frequency distribution per crop number of records in the knowledge base. The distribution is observed to be exponential in nature, i.e. there are fewer records in the knowledge base regarding a large number of crops, and a small set of crops dominates the knowledge base. Figure 14 shows the pie chart of the top 24 crops acquiring $\approx 51\%$ of the knowledge base. Moreover, it was found that most of the crops for which the models couldn’t revert any answers are among the least-interested crops (crops acquiring $\approx 1.05\%$ of the knowledge base).

To understand the model’s response times, figure 15 shows the frequency distribution of the number of solutions retrieved from the models against the response-time bins. The figure shows that among all the models, RRM1 has the least number of responses with the lowest response-retrieval time (RRT) bin. The maximum RRT obtained by RRM1 is 507.82 secs; therefore, the frequency distribution of RRM1’s RRT is observed to be stretched. Moreover, the frequency distribution corresponding to RRM2 is very short (maximum RRT is 82.97 secs), as most questions get answered in smaller RRT bins. Furthermore, the distribution corresponding to RRM3 is similar to that of RRM2 (maximum RRT of RRM3 is 82.10 secs). RRM3 has a slightly higher number of answers in the smaller RRT bins from 1.5 sec to 11.5 sec, which leads to lesser RRT on average.

It is found that a large number of crops in the base are present with a small number of records, whereas a few crops dominate the base. Moreover, it is observed that the wrong answers reverted by the models belong to the crops with fewer

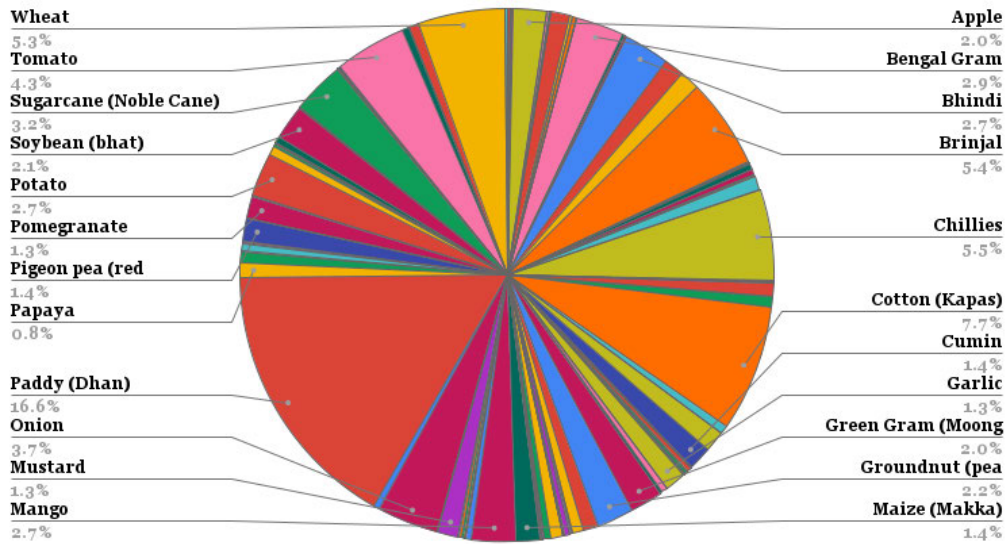


FIGURE 14. Percentages of the major crops in the knowledge base.

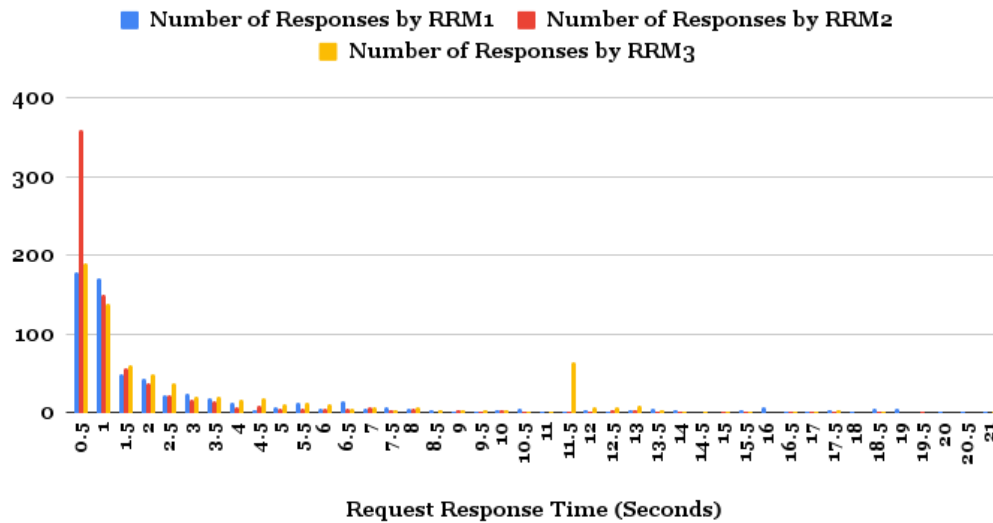


FIGURE 15. Frequency distribution of the response-retrieval time.

records in the knowledge base. A similar distribution was observed regarding the models’ RRT; it was noted that many questions get answered with shorter RRT, whereas queries related to the “famous” crops take longer.

The proposed system in our research stands out with an impressive accuracy rate of 98.50% (RRM1), showcasing its exceptional performance in automated query-response systems for agricultural decision-making. Other notable systems have been evaluated, including the “KisanQRS”, a query-response generation system built with the KCC dataset, with an accuracy of 97.12% [49]. Moreover, the model proposed by Kumar et al. [50] achieves an accuracy of 96.10%, on par with “Krushni-The Farmer Chatbot” [51]. Lastly, “AgriBot”, an intelligent interactive interface designed to assist farmers, exhibits a still respectable accuracy

of 78% [52]. These results highlight the diverse landscape of systems designed to support agricultural decision-making, with the proposed RRM1 leading the way in accuracy and performance.

In our research, we’ve focused on accuracy and considered an essential aspect that sets us apart from existing research - the Average Response Retrieval Time. While other studies may primarily concentrate on accuracy and performance metrics, our work strongly emphasises evaluating how quickly the system can retrieve responses, ensuring that it not only provides accurate answers but does so promptly. This emphasis on response time is critical in practical applications, especially in agricultural decision-making.

While our research presents several strengths, it’s important to acknowledge certain limitations. First, the study

primarily focuses on the agricultural context and may not be directly applicable to other domains. Second, the quality of the data input can influence the system's performance. Lastly, while we've considered Average Response Retrieval Time, further optimization for even quicker responses may be required for highly time-sensitive situations.

VI. CONCLUSION

With the global increase in ICT infrastructure, the agriculture sector needs systems to cope with the farmers' demand for help. Moreover, agriculture-related helplines and support centres are currently lacking expertise [7]. Delays in the experts' advice may have a serious impact on farmers' lives as well as the national economy. For this situation, we propose AgriResponse, a framework to provide text-based plant-protection-related solutions to Indian farmers in real-time. The helpline operators can also use the proposed framework as a second opinion to the experts' advice.

Challenges while designing the proposed framework include creating a knowledge base for questions and answers covering various crops grown in India. Secondly, developing smart query-response retrieval systems for better accuracy and response-retrieval time. As a solution, a new way of constructing a knowledge base has been reported, using the call-log records of the farmers' helpline centres. Moreover, three separate models (RRM1, RRM2, and RRM3) are developed with different searching capabilities for response retrieval. The models are validated on a question bank consisting of 755 queries covering 151 crops. Models' performances are evaluated using three metrics (AP, CWPS, and ARRT) on the question bank. The experiments found that RRM1 has the highest AP and CWPS (87.41% and 98.5%, respectively), with the lowest performance in terms of ARRT (11.60 sec). RRM2's performance is noted to be the lowest in terms of AP and CWPS (70.59% and 92.91% respectively), and best in terms of ARRT (2.29 sec). Whereas RRM3's performance was comparable to the best models in their respective metrics (AP = 85.43%, CWPS = 97.15%, and ARRT = 4.18 sec). Therefore, it can be stated that RRM3 is a good choice of response-retrieval model for our objective. For future work, we intend to merge knowledge bases from multiple sources (QA forums, blogs, articles, etc.) to increase the AP and CWPS of the system. Moreover, we intend to incorporate advanced string-matching algorithms integrated with machine learning for better framework performance.

CONFLICTS OF INTEREST

The author(s) declare(s) that there is no conflict of interest.

DATA AVAILABILITY STATEMENT

The dataset used in the current research study is publicly available.

REFERENCES

- [1] T. Wang, Z. Wang, L. Guo, J. Zhang, W. Li, H. He, R. Zong, D. Wang, Z. Jia, and Y. Wen, "Experiences and challenges of agricultural development in an artificial oasis: A review," *Agricult. Syst.*, vol. 193, 2021, Art. no. 103220.
- [2] Kisaan Knowledge Management System Gpvernment of India. (2021). *Kisan Call Center*. [Online]. Available: <https://dackkms.gov.in/>
- [3] P. Jaisridhar, "Impact of Kisan call center on technological adoption among dairy farmers of Tamilnadu," Ph.D. thesis, NDRI, Karnal, Karnal, India, 2013.
- [4] S. Kavitha and A. Nallusamy, "A study on socio economics characteristics of Kisan call centre beneficiaries and non-beneficiaries in mahaboobnagar district of telangana," *J. Pharmacognosy Phytochem.*, vol. 8, no. 3, pp. 4660–4663, 2019.
- [5] K. Chachra, G. Seelam, H. Singh, M. Sarkar, A. Jain, and A. Jain, "The impact of Kisan call centers on the farming sector," in *Environmental and Agricultural Informatics: Concepts, Methodologies, Tools, and Applications*. Hershey, PA, USA: IGI Global, 2020, pp. 66–78.
- [6] Department of Agriculture and Farmers Welfare. (2020). *Kisan Call Centre*. [Online]. Available: <https://agricoop.nic.in/sites/default/files/KCC%20WEBSITE.pdf>
- [7] P. Ajawan, P. Desai, and V. Desai, "Smart Sampark—An approach towards building a responsive system for kisan call center," in *Proc. IEEE Bengaluru Humanitarian Technol. Conf. (B-HTC)*, Oct. 2020, pp. 1–5.
- [8] National Informatics Centre. (2021). *Open Government Data Platform India*. [Online]. Available: <https://data.gov.in/>
- [9] S. Godara. (2021). *Agriresponse Online Repository*. [Online]. Available: <https://github.com/Samarth-Godara/AgriResponse>
- [10] S. Godara. (Oct. 2021). *Agriresponse: A Query-Response Generation Model for Indian Farmers*. [Online]. Available: <https://www.codeocean.com/>
- [11] S. K. Dwivedi and V. Singh, "Research and reviews in question answering system," *Proc. Technol.*, vol. 10, pp. 417–424, Jan. 2013.
- [12] B. F. Green, A. K. Wolf, C. Chomsky, and K. Laughery, "Baseball: An automatic question-answerer," in *Proc. Papers Presented Western Joint IRE-AIEE-ACM Comput. Conf. IRE-AIEE-ACM (Western)*, 1961, pp. 219–224.
- [13] W. A. Woods, "Progress in natural language understanding: An application to lunar geology," in *Proc. June 4–8, Nat. Comput. Conf. Expo. (AFIPS)*, 1973, pp. 441–450.
- [14] D. G. Bobrow, R. M. Kaplan, M. Kay, D. A. Norman, H. Thompson, and T. Winograd, "GUS, a frame-driven dialog system," *Artif. Intell.*, vol. 8, no. 2, pp. 155–173, 1977.
- [15] P. Clark, J. Thompson, and B. Porter, "A knowledge-based approach to question-answering," in *Proc. AAAI*, vol. 99, 1999, pp. 43–51.
- [16] B. Katz, "Annotating the world wide web using natural language," in *Proc. RIAO*, 1997, pp. 136–159.
- [17] H. Chung, Y.-I. Song, K.-S. Han, D.-S. Yoon, J.-Y. Lee, H. C. Rim, and S.-H. Kim, "A practical QA system in restricted domains," in *Proc. Conf. Question Answering Restricted Domains*, 2004, pp. 39–45.
- [18] A. Mishra, N. Mishra, and A. Agrawal, "Context-aware restricted geographical domain question answering system," in *Proc. Int. Conf. Comput. Intell. Commun. Netw.*, Nov. 2010, pp. 548–553.
- [19] P. Moreda, H. Llorens, E. Saquete, and M. Palomar, "Combining semantic information in question answering systems," *Inf. Process. Manage.*, vol. 47, no. 6, pp. 870–885, Nov. 2011.
- [20] D. Hristovski, D. Dinevski, A. Kastrin, and T. C. Rindflesch, "Biomedical question answering using semantic relations," *BMC Bioinf.*, vol. 16, no. 1, pp. 1–14, Dec. 2015.
- [21] X. Huang, B. Wei, and Y. Zhang, "Automatic question-answering based on Wikipedia data extraction," in *Proc. 10th Int. Conf. Intell. Syst. Knowl. Eng. (ISKE)*, Nov. 2015, pp. 314–317.
- [22] X. Xie, W. Song, L. Liu, C. Du, and H. Wang, "Research and implementation of automatic question answering system based on ontology," in *Proc. 27th Chin. Control Decis. Conf. (CCDC)*, May 2015, pp. 1366–1370.
- [23] F. Colas and P. Brazdil, "Comparison of SVM and some older classification algorithms in text classification tasks," in *Proc. IFIP Int. Conf. Artif. Intell. Theory Pract.* Cham, Switzerland: Springer, 2006, pp. 169–178.
- [24] A. Ittycheriah, "IBM's statistical question answering system," in *Proc. TREC*, 2001, pp. 1–6.

- [25] A. Moschitti, "Answer filtering via text categorization in question answering systems," in *Proc. 15th IEEE Int. Conf. Tools with Artif. Intell.*, 2003, pp. 241–248.
- [26] K. Zhang and J. Zhao, "A Chinese question-answering system with question classification and answer clustering," in *Proc. 7th Int. Conf. Fuzzy Syst. Knowl. Discovery*, vol. 6, Aug. 2010, pp. 2692–2696.
- [27] L. Han, Z.-T. Yu, Y.-X. Qiu, X.-Y. Meng, J.-Y. Guo, and S.-T. Si, "Research on passage retrieval using domain knowledge in Chinese question answering system," in *Proc. Int. Conf. Mach. Learn. Cybern.*, Jul. 2008, pp. 2603–2606.
- [28] R. Soricut and E. Brill, "Automatic question answering using the web: Beyond the factoid," *Inf. Retr.*, vol. 9, no. 2, pp. 191–206, Mar. 2006.
- [29] R. Higashinaka and H. Isozaki, "Corpus-based question answering for why-questions," in *Proc. 3rd Int. Joint Conf. Natural Lang. Process.*, vol. 1, 2008, pp. 418–425.
- [30] J. Fu, J. Xu, and K. Jia, "Domain ontology based automatic question answering," in *Proc. Int. Conf. Comput. Eng. Technol.*, Jan. 2009, pp. 346–349.
- [31] H. Toba, Z.-Y. Ming, M. Adriani, and T.-S. Chua, "Discovering high quality answers in community question answering archives using a hierarchy of classifiers," *Inf. Sci.*, vol. 261, pp. 101–115, Mar. 2014.
- [32] M. Medved and A. Horák, "AQA: Automatic question answering system for Czech," in *Proc. Int. Conf. Text, Speech, Dialogue*. Cham, Switzerland: Springer, 2016, pp. 270–278.
- [33] E. Lima, W. Shi, X. Liu, and Q. Yu, "Integrating multi-level tag recommendation with external knowledge bases for automatic question answering," *ACM Trans. Internet Technol.*, vol. 19, no. 3, pp. 1–22, Aug. 2019.
- [34] L. A. Ha and V. Yaneva, "Automatic question answering for medical MCQs: Can it go further than information retrieval?" in *Proc. Natural Lang. Process. Deep Learn. World*, Oct. 2019, pp. 1–5.
- [35] D. Oniani and Y. Wang, "A qualitative evaluation of language models on automatic question-answering for COVID-19," in *Proc. 11th ACM Int. Conf. Bioinf., Comput. Biol. Health Informat.*, Sep. 2020, pp. 1–9.
- [36] S. G. Aithal, A. B. Rao, and S. Singh, "Automatic question-answer pairs generation and question similarity mechanism in question answering system," *Int. J. Speech Technol.*, vol. 51, no. 11, pp. 8484–8497, Nov. 2021.
- [37] D. Ravichandran and E. Hovy, "Learning surface text patterns for a question answering system," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2001, pp. 41–47.
- [38] D. Zhang and W. S. Lee, "Web based pattern mining and matching approach to question answering," in *Proc. TREC*, vol. 2, 2002, p. 497.
- [39] M. A. Greenwood and R. Gaizauskas, "Using a named entity tagger to generalise surface matching text patterns for question answering," in *Proc. Workshop Natural Lang. Process. Question Answering (EACL)*, 2003, pp. 29–34.
- [40] H. Cui, M.-Y. Kan, and T.-S. Chua, "Soft pattern matching models for definitional question answering," *ACM Trans. Inf. Syst.*, vol. 25, no. 2, p. 8, Apr. 2007.
- [41] A. K. Saxena, G. V. Sambhu, S. Kaushik, and L. V. Subramaniam, "IITD-IBMIRL system for question answering using pattern matching, semantic type and semantic category recognition," in *Proc. TREC*, 2007, pp. 1–9.
- [42] V. K. Viswanath, C. G. V. Madhuri, C. Raviteja, S. Saravanan, and M. Venugopalan, "Hadoop and natural language processing based analysis on kisan call center (KCC) data," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2018, pp. 1142–1151.
- [43] S. Godara and D. Toshniwal, "Sequential pattern mining combined multi-criteria decision-making for farmers' queries characterization," *Comput. Electron. Agricult.*, vol. 173, Jun. 2020, Art. no. 105448.
- [44] S. Godara, D. Toshniwal, R. Parsad, R. S. Bana, D. Singh, J. Bedi, A. Jhajhria, J. P. S. Dabas, and S. Marwaha, "Agrimine: A deep learning integrated spatio-temporal analytics framework for diagnosing nationwide agricultural issues using farmers' helpline data," *Comput. Electron. Agricult.*, vol. 201, Oct. 2022, Art. no. 107308.
- [45] S. Godara and D. Toshniwal, "Deep learning-based query-count forecasting using farmers' helpline data," *Comput. Electron. Agricult.*, vol. 196, May 2022, Art. no. 106875.
- [46] S. Godara, D. Toshniwal, R. S. Bana, D. Singh, J. Bedi, R. Parsad, J. P. S. Dabas, A. Jhajhria, S. Godara, R. Kumar, and S. Marwaha, "AgrIntel: Spatio-temporal profiling of nationwide plant-protection problems using helpline data," *Eng. Appl. Artif. Intell.*, vol. 117, Jan. 2023, Art. no. 105555.
- [47] S. K. Mohapatra and A. Upadhyay, "Using TF-IDF on Kisan call centre dataset for obtaining query answers," in *Proc. Int. Conf. Commun., Comput. Internet Things (IC3IoT)*, Feb. 2018, pp. 479–482.
- [48] D. Diefenbach, V. Lopez, K. Singh, and P. Maret, "Core techniques of question answering systems over knowledge bases: A survey," *Knowl. Inf. Syst.*, vol. 55, no. 3, pp. 529–569, Jun. 2018.
- [49] M. Z. U. Rehman, D. Raghuvanshi, and N. Kumar, "KisanQRS: A deep learning-based automated query-response system for agricultural decision-making," *Comput. Electron. Agricult.*, vol. 213, Oct. 2023, Art. no. 108180.
- [50] M. Kumar, K. K. Chaturvedi, A. Sharma, A. Arora, M. S. Farooqi, S. B. Lal, A. Lama, and R. Ranjan, "An algorithm for automatic text annotation for named entity recognition using Spacy framework," ICAR, Delhi, India, Tech. Rep., 2023.
- [51] M. Momaya, A. Khanna, J. Sadavarte, and M. Sankhe, "Krushi—The farmer chatbot," in *Proc. Int. Conf. Commun. Inf. Comput. Technol. (ICCICT)*, Jun. 2021, pp. 1–6.
- [52] D. Sawant, A. Jaiswal, J. Singh, and P. Shah, "AgriBot—An intelligent interactive interface to assist farmers in agricultural activities," in *Proc. IEEE Bombay Sect. Signature Conf. (IBSSC)*, Jul. 2019, pp. 1–6.



SAMARTH GODARA received the M.Tech. degree from the National Institute of Technology, Jalandhar, Punjab, India, and the Ph.D. degree specializing in artificial intelligence from IIT Roorkee. He contributes to the scientific community as a Scientist with the ICAR-Indian Agricultural Statistics Research Institute, New Delhi, India. He has authored numerous research papers in big data analytics and deep learning. His research interests include artificial intelligence, machine learning, and data analytics.



JATIN BEDI received the Ph.D. degree from IIT Roorkee, which made him a prominent figure in advanced technology and data analysis. He is an Assistant Professor with the Thapar Institute of Engineering and Technology, specializes in artificial intelligence, machine learning, and data analytics. He has authored numerous high-impact research articles in the fields of deep learning and time series forecasting, which have published in prestigious international journals.



RAJENDER PARSAD received the Ph.D. degree in agricultural statistics. Currently, he is the Director of the ICAR-Indian Agricultural Statistics Research Institute, New Delhi. He is a highly accomplished scientist with a profound academic journey. He was honoured with the NAAS Recognition Award for his significant contributions to the field of social sciences, in 2015 and 2016. In addition, he received the National Award in Statistics, for his outstanding work in statistics, in 2010 and 2011; and the Prof. P. V. Sukhatme Gold Medal Award from the Indian Society of Agricultural Statistics, in 2010. He held the ICAR National Fellow, recognizing his remarkable achievements in the field of agriculture and statistics, from January 2005 to April 2009. He is renowned for his contributions to agricultural statistics, design of experiments, sampling techniques, and statistical computing.



DEEPAK SINGH received the Ph.D. degree from Amity University, New Delhi. He is a dedicated scientist, specializing in statistical analysis, machine learning, and data analytics. He is affiliated with the ICAR-Indian Agricultural Statistics Research Institute, New Delhi, India. His work focuses on leveraging data-driven insights to advance agricultural research and decision-making. His expertise is underscored by his multiple publications in esteemed international journals and highlighting his commitment to advancing the field of agriculture through rigorous research and data-driven insights.



RAM SWAROOP BANA received the M.Sc. and Ph.D. degrees from the Indian Agricultural Research Institute (IARI), New Delhi, India. He is a distinguished senior scientist with a wealth of expertise in the field of agronomy and vegetable science. His dedication to the study of agriculture has made him an invaluable contributor to the institute's mission. With an extensive background in agronomy, he brings a wealth of knowledge and experience to the research community. His distinguished academic background is complemented by a prolific research portfolio, encompassing over 50 research articles published in renowned international journals.



SUDEEP MARWAHA received the B.Sc. degree in electronics from the University of Delhi, New Delhi, India, in 1995, the M.Sc. degree in computer applications from the Indian Agricultural Research Institute, New Delhi, and the Ph.D. degree in computer science from the University of Delhi, in 2008. He is a Principal Scientist and a Professor with the Division of Computer Applications, ICAR-IASRI, New Delhi. He completed the following projects: Solution Architect for Semantic Web-Enabled Systems, Management Information Systems, ERP (Oracle Apps), Knowledge Base Systems, and Image Analysis-Based Systems. He has published more than 50 research articles. His research interests include artificial intelligence, semantic web, and ontologies.

...