AGRICULTURAL RELATED QUERY CLARIFIER SYSTEM USING BERT MODEL

A PROJECT REPORT

Submitted by

VIJAI KUMAR S

(2020178060)

submitted to the Faculty of

INFORMATION AND COMMUNICATION ENGINEERING

in partial fulfillment for the award of the degree

of

MASTER OF COMPUTER APPLICATIONS

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING, GUINDY

ANNA UNIVERSITY

CHENNAI 600 025

JULY 2022

ANNA UNIVERSITY

CHENNAI - 600 025

BONA FIDE CERTIFICATE

Certified that this project report titled AGRICULTURAL RELATED QUERY CLARIFIER SYSTEM USING BERT MODEL is the bonafide work of Mr.VIJAI KUMAR S who carried out project work under my supervision. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on this or any other candidate.

PLACE: Chennai
DATE:

Dr. D.NARASHIMAN
TEACHING FELLOW
PROJECT GUIDE
DEPARTMENT OF IST, CEG
ANNA UNIVERSITY
CHENNAI 600025

COUNTERSIGNED

Dr. S.SRIDHAR
HEAD OF THE DEPARTMENT
DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING, GUINDY
ANNA UNIVERSITY
CHENNAI 600025

ABSTRACT

In India, agriculture is the foundation of the economy, and good knowledge of agricultural practices is the key to optimal agricultural growth and production. This agricultural query clarifier system was built based on the dataset from Kisan Call Center, which is used to answer questions of the farmers.This project is ideally suited to the Natural Language Processing domain. This system is robust enough to answer queries related to market rates, plant protection and government schemes. Access to this system is available 24x7, it is available through any electronic device, and the information is delivered in an easy to understand manner. This system is based on a Bert model. An integrated system like this would allow farmers to advance towards easier information regarding farming related practices, leading to better agri output. The job of the KCC workforce would be made easier and the difficult work of various such workers can be redirected to a better goal.

ACKNOWLEDGEMENT

I would like to express my profound sense of gratitude to my guide **DR.D.NARASHIMAN** Teaching Fellow, Department of Information Science and Technology Anna University, for his invaluable support, guidance in project and constant encouragement for the successful completion of this project.He gave me the courage to complete this project and supported me in my design decisions of the project.

I would like to express my gratitude to **DR. S.SRIDHAR** Professor, Head of the Department, Information Science and Technology, Anna University, for his kind support and for providing necessary facilities to carry out the work.

I like to express my sincere thanks to the project committee members **DR.R.GEETHA RAMANI** Professor, **DR.S.SENDHIL KUMAR** Associate Professor, **DR.B.SENTHIL NAYAKI** Teaching Fellow, **MR.H.RIASUDHEN** Teaching Fellow, Department of Information Science and Technology, Anna University, Chennai for their valuable guidance and technical support critical reviews throughout the course of my project.

I wish to thank all the staff of Department of Information Science and Technology, Anna University, for their technical support for this project.

I am grate-full for the support and encouragement I've received from my parents and friends.

VIJAI KUMAR S

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

KCC	Kisan Call Center
NLP	Natural Language Processing
TRAI	Telecom Regulatory Authority of India
CSV	comma-separated values
DEO	Data Entry Operator
GDP	Gross Domestic Product
AI	Artificial Intelligence
QA	Question Answer

CHAPTER 1

INTRODUCTION

Agriculture plays a significant role in economic development in India, contributing approximately 15.4% to overall Gross Domestic Product[GDP] and employing roughly 55% of the population. However, most farmers do not have access to authentic information about the latest farming practices and trends.In part, this is due to the fact that people in the farming industry take a longer time to adopt new technologies.

Farmers are traditionally trained and consulted by field officers who visit their farms from time to time. It is often difficult to find information or contact officials in rural villages because they are inaccessible. The result is that farmers often do not have access to agricultural information that can help them make better decisions regarding the crops they cultivate. This reduces crop yields, and reduces market efficiency. A farmer's earnings, time, and opportunities to increase crop yields are severely affected by these factors.

As of Mar 2021, the number of urban mobile subscribers was 645.20 million, while the number of rural subscribers was 535.75 million, approximately 45% of Indian internet users will be from the rural sectors according to Telecom Regulatory Authority of India[TRAI][1]. It is difficult for the government to spread vital information about farming. The problem also worsens due to the spread of misinformation. Because of the vast language diversity and a lack of confidence in modern technology, these problems exist. It seems promising to use digital devices to spread agriculture-related information in such a scenario.

1.1 SCOPE OF THE PROJECT

In the Agricultural sector millions of people involved directly as well as indirectly. At the same time Indian agricultural sector do very slow growth on technology side. Farmers can't able to get an authentic information whenever they needed. This could lead to many losses to farmer. Our project helps farmer to clarify their doubts 24*7. Authentic information is will be provided with by using Kissan Call Center[KCC] datasets.

1.2 MOTIVATION OF THE PROJECT

Majority of the farmers in our country can't able to get an authentic information when ever they needed. Its difficult to rely on government authorities all the time, and also not many experts are available at all times. This leads to the development of the system which clarify farmers query whenever they want (24*7).This could also reduce the work load of government officials. Query which is provided by the system also an authentic information from Government call center KCC.

1.3 OVERVIEW OF THE PROJECT

It aims to develop a system that gives authentic answers to farmers, 24 hours a day, 7 days a week, whenever they have agricultural related questions. And also there is no limit to how many times they can access the information.

1.4 OUTLINE OF THE REPORT

The rest of the project is organized as follow:

Chapter 2: This Chapter Describes about the Literature Survey done for the project

Chapter 3: This Chapter Illustrates the system design, system architecture and module description of the proposed method.

Chapter 4: This Chapter going to Describes the Implementation and Results details

Chapter 5: This Chapter Shows the Conclusion and Future work.

CHAPTER 2

LITERATURE SURVEY

2.1 INTRODUCTION

The process of identifying the appropriate representation based on the certain conditions and the characteristic features of the representing elements also Machine Learning and Natural Language Processing concept related to this project are discussed here.

2.2 RELATED WORKS

The Government has implemented various measures for agriculture related IT services which provide access to a central knowledge bank in order to address the above defined problem. The most prominent services are listed below.

Agri App is one of the most popular apps among farmers. It has a rating of 4.2 out of 5 on the Google Play store. This portal brings information about farming resources and government services through an online mobile application to the farmers. It also provides a chat option for farmers which enables them to chat with an agri expert using this app efficiently. However, AgriApp is a knowledge bank wherein the user has to search for a particular piece of information manually and if the user opts to chat with the application operator instead of searching manually, the user has to wait for a significant period of time for a response from the operator.

Farmers Portal makes use of the Internet to make knowledge is accessible to farmers. There is a lot of information related to farming on this website, but it's mainly presented in English and Hindi. Despite this, most farmers are not literate enough to operate modern devices properly, which poses a significant challenge to this service.[2]

Kissan Call Center [KCC][3] is a helpline service for farmers to clarify their queries over the phone. Since the service facilitates a telephonic conversation, this service is able to cater to the needs of farmers on an individual basis as the information is provided in their native language and relevant to their location. Additionally, the farmers get valuable information related to new farming practices. This service reduces the difficulty of the farmer to ask for help related to latest agricultural practices which also helps in building the trust of the rural class on the Government. However, KCC services are only available from 6 AM to 10 PM, and skilled labor with good knowledge of agricultural practices is required to operate the Call Center. Also, it is observed that with time, queries to KCC have increased exponentially due to increase in awareness among farmers as well as technology adoption. This has the potential to generate the need to set up new call centers which will require massive cost along with training the human resource.

According to the analysis of Kisan Call Center data[KCC], about 4.8 million calls were made to KCC in 2018-2019 which increased to about 5.47 million calls in 2020-2021. This shows a 11% increase in calls from 2019 to 2021. only 5% unique new queries were made in 2021 compared to previous year. The number of questions are increasing gradually, and soon these call centers may not be able to efficiently answer all these queries on time, plus most of the queries are redundant. Hence, a scalable solution is needed to accommodate the increase in the number of queries in a better way. This project use the power of Artificial Intelligence[AI] to build a solution to this problem.

2.3 LIMITATION OF EXISTING WORKS

There exists a good number of Question and Answers[QA] models which deal with a similar problem. G.Ifri,etc are[4] use a knowledge graph based in method, where the knowledge graph is built upon the data and questions are answered using the knowledge graph. Another work carried out by Robin Jia Pajpurkar,etc [5] is a comprehension based question answering system. In this systems, for every question the system generates an answer based on the knowledge gathered by understanding the comprehensions. However, these methods cannot be used to solve the answer ranking problem, because neither our data is properly formatted in a comprehension nor the facts can be extracted to form a knowledge graph. Another way to solve the problem is using Question Answer pair hashing. However, it must be modified to fit the needs because many semantically similar questions have different answers.

CHAPTER 3

SYSTEM DESIGN

The overall system architecture, individual module description and requirement specification are explained in the following section. These details help understanding the concept and complexity involved in a project and in each module.

3.1 SYSTEM ARCHITECTURE

The System Architecture diagram provides necessary information about the workflow of the project. The module diagram shows the functionalities that are done in that module. Figure 3.1 diagram of proposed system.

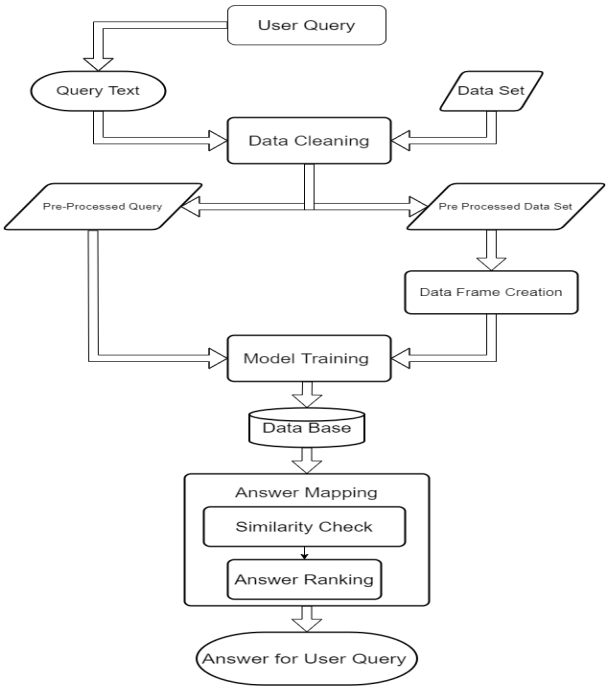

Figure 3.1: System Architecture

3.2 MODULE DESCRIPTION

Each and every module deployed for this project is explained clearly and elaborately in this section.

3.2.1 DATASET GENERATION

In this module, KCC data is collected in Java Script Object Notation [JSON] format in this Website https://data.gov.in[3][6] Indian government owned. Through this process, I developed a code to download JSON data from KCC center which downloads the data automatically without any human intervention and converts it into Comma Separated Values[CSV] format using Python libraries. Basically, CSV is a simple file format which is used to store tabular data (number and text) such as a spreadsheet in plain text. A CSV file would be more convenient for this specific NLP project.

CHALLENGES: Apart from collecting the data, there were three significant challenges. First, we saw a lack of consistency in the format of the questions and answers. Most of the data is poorly written with many redundant words, spelling errors, incorrect grammar and punctuation. These features make the process of information extraction from the data a difficult task. The questions are well written compared to the answers in terms of the ease of understanding. Hence, we chose to process the questions to find the critical words. Also, the answers are very vague and are not framed sentences. Various answers are just numbers. Processing answers to understand their meaning and relevance to a particular input question is a challenging task.

And some of the queries are registered in the regional language which poses a problem in pre-processing the data because the translation resources for specific languages are limited. Various questions and answers uses a few or all

words from a native language. Also, most of them are not proper sentences. Such quality of data makes it difficult to process them even after translation to English.

In order to check the accuracy of our system, we need a dataset with ground truth corresponding to each query which does not exist. Lack of truth values made it difficult to determine if the answer given as output for a given input query to the system is correct or not. Such a metric is necessary to measure the reliability of our model. Hence, the determination of a suitable metric for the model was a significant task.

Algorithm 3.1 Algorithm:Data-Set Generation

```
Initialize the values distCode,stateCode,month,year,num,
create csv file
pass the value in data.gov Database
if values == true then
    temp = Json data as per values
else
    increase the values
end if
if temp > 0 then
    append JSON Data in existing CSV file.
    increase the values
else
    increase the values
end if
```

3.2.2 DATA CLEANING

The process of data cleaning[7] is essential in any machine learning model, but it is especially necessary for Natural Language Processing.Without the cleaning process, a dataset is often a cluster of words that computers are unable to understand. The steps involved in cleaning data in a machine learning text pipeline for this project will be discussed here.Figure 3.2 shows the Data Cleaning module

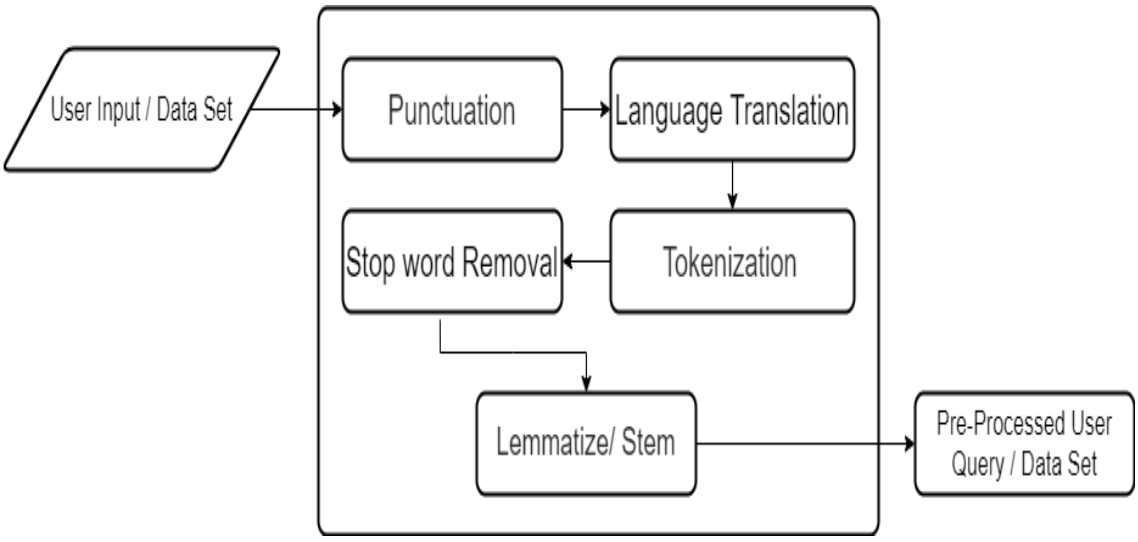

Figure 3.2: Data Cleaning Module

Punctuation: Text processing techniques include removing punctuation from the textual data. The punctuation removal process will help to treat each text equally. For example, the word farmer and farmer! are treated equally after the process of removal of punctuation's. The data we collected from KCC have lot punctuation's. We use this punctuation removal mainly in two columns "Query Answer for the Query".

Language translation: According to KCC data, it contains many multiple Indian languages. So we converted the dataset to English language. We can perform our model training more easily by converting these text files to English.

Tokenization and lower casing: Tokenization is breaking the raw text into small chunks. Tokenization breaks the raw text into words, sentences called tokens. These tokens help in understanding the context or developing the model for NLP. The tokenization helps in interpreting the meaning of the text by analyzing the sequence of the words. This process converting the Sentence to list. List of text also converts text to lowercase to maintain uniformity while training models, for example, Potato, Potato, PoTAto are renamed as "potato".

Stop Word Removal: Stop Word Removal technique is used in this module, Stop Word is commonly used word in a sentence,example: "such", "a", "or",etc.

After tokenization we remove those common word from the list because we don't want these word to take up space in our database, or taking up our valuable processing time. For this we can remove them, by storing a list of word that we consider to stop words.

Stemming: Stemming is the process of removing a part of a word, or reducing a word to its stem or root. This means we're reducing a word to its dictionary root. This technique used to minimizes the word length to be trained. For example in our data set this word is used so many time "Asked", "Asking", "Ask" are stemmed to its root word as "Ask".

Algorithm 3.2 Algorithm:Data Cleaning

```
df = get dataset.csv
pass df , perform punctuation
for <df ← 0 to lengthofdf> do
    punctuation = '!"()*+,-./:;<=>?@[]^|
    <df[i].remove punctuation>
end for
pass df , perform Language Translation
for <df ← 0 to lengthofdf> do
    <df[i].convert to english>
end for
pass df , perform Tokenization
for <df ← 0 to lengthofdf> do
    <df[i].Tokanize. to lower case>
end for
pass df , perform stop word removal
for <df ← 0 to lengthofdf> do
    if df == stopword then
        df = remove stop word
    else
        do nothing
    end if
end for
pass df , perform stemming
for <df ← 0 to lengthofdf> do
    <df[i].find root>
    change df[i] = rootword
end for
```

3.2.3 DATA FRAME CREATION

There are several data columns in the preprocessed data set. There are many columns with null values. In order to obtain an exact answer, a data frame has been constructed by considering relevant information. In order to avoid redundancy, used to group all answers to a particular question into a list. Finally, the data-frame containing the query, query-type, state, district, time of query and the list of answers corresponding to that query is given as an input into our model.Figure 3.3 shows the Data Frame Creation module

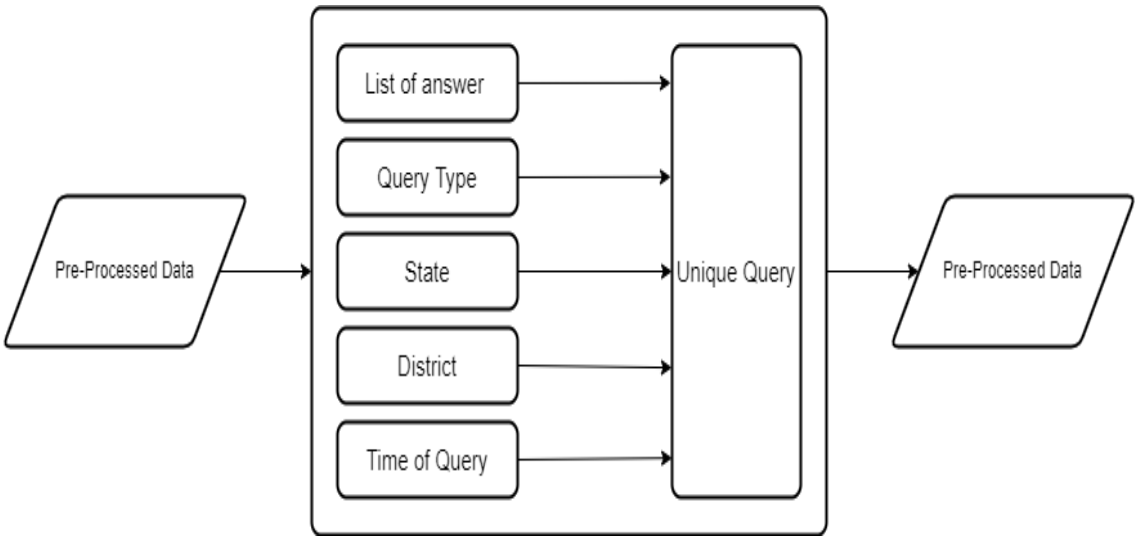

Figure 3.3: Data Frame Creation

Algorithm 3.3 Algorithm:Data Frame Creation

1: get $pre-processeddataset$ as df
2: dict =
3: **for** $<df \leftarrow 0$ `to` $lengthofdf>$ **do**
4: **if** $df[i]$ == $unique$ or $df[i]$ == $alreadyexisist$ **then**
5: dictkey:df[i] = Query type,State, District, Time of Query and List of Answers
6: **end if**
7: **end for**
8: dataframe = dict

3.2.4 MODEL TRAINING

The data frame created in the previous module is used to train the model. The Sen2Vec model can be described as a method of converting a sentence into a vector, where the allotted weight to each dimension of the vector represents its inclination towards a particular context. The primary purpose of this model is to cluster the similar sentences without taking into consideration the ordering of the words.

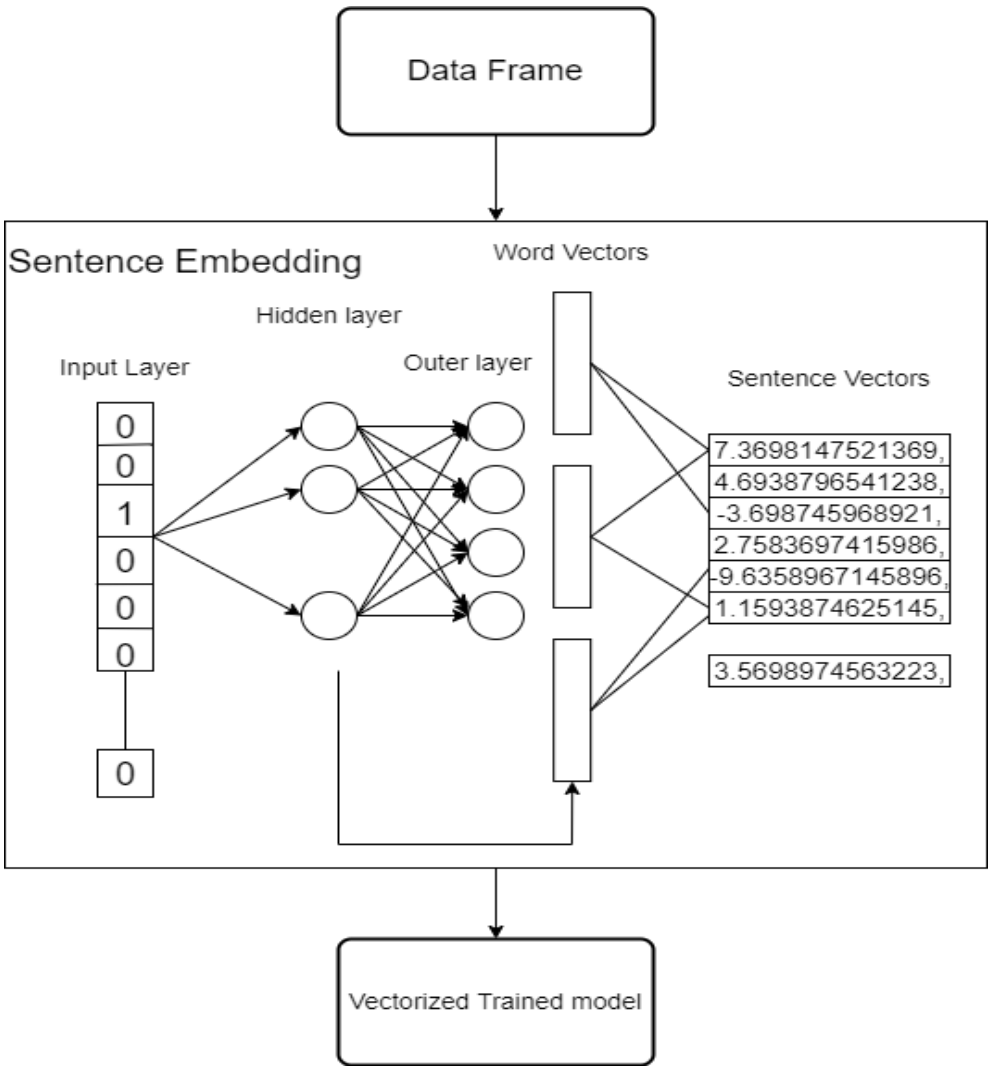

Figure 3.4: Model Training

Considering the improper format of the queries, we attempt to match input question to questions which are present in our given dataset rather than processing the answers - the idea being that given the size of the dataset and redundancy, the question is highly likely to be already present. We divided the collected data into two parts - train and test. Using the training data, we train our model based on Sen2Vec[8] and then for each query in test data we find the most similar question indexed in the training data.

This model is trained using only Query from Data Frame. An input to this model is the question, and it is tokenized before being analyzed. Each word is converted to an vector after all are tokenized. Using the Sentence Embedding method word vectors are converted into Sentence Vectors. Figure 3.4 shows the Model Training.

Algorithm 3.4 Algorithm:Model Training

1: get $dataferame$ as df
2: model = $bert-base-nli-mean-tokens$
3: querys = []
4: **for** <$df \leftarrow 0$ `to` $lengthofdf$> **do** querys[i] = df[i].keys()
5: **end for**
6: trained model = $model.train[querys]$

3.2.5 ANSWER MAPPING

An analysis of similarity is performed based on the training model and the vectorized user query text. To measure similarity between Vectorized user query and Trained model, we use cosine similarity[9]. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction.In this way, we can determine which matrix is most similar in the trained model based on the vectorized query. This will provide us with a list of answers to the particular question

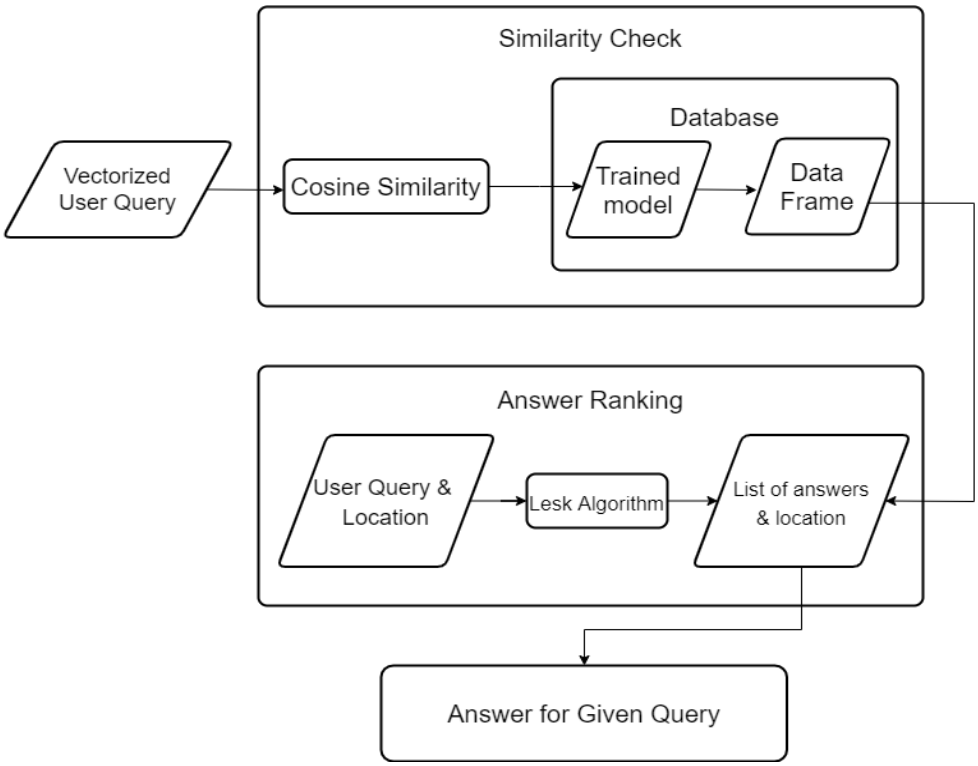

Figure 3.5: Answer Mapping

Word Sense Disambiguation basically solves the ambiguity that arises in determining the meaning of the same word used in different situations. Lesk Algorithm is a classical Word Sense Disambiguation algorithm[10]. The Lesk algorithm is based on the idea that words in a given region of the text will have a similar meaning. In the Simplified Lesk Algorithm, the correct meaning of each word context is found by getting the sense which overlaps the most among the given context and its dictionary meaning. We rank the answer according to the Lesk Algorithm based on the location of the User and the existing location in the list of answers. A ranking method[4] is used in order to determine the answer for a particular query.Figure 3.5 shows the Model Training.

Algorithm 3.5 Algorithm:Answer Mapping

1: get user query as q
2: find $vector$ for q
3: queryTrained = model.train[q]
4: maxSimilarity = cosineSimilarity(queryTrained and trainedmodel)
5: predictedQuery is query of index$maxSimilarity$
6: ListOfAnswer = querys of predictedQuery
7: Answer = leskAlgorithm of $ListOfAnswer$,$predictedQuery$

3.2.6 DATA ANALYSIS

To understand our dataset, we explored the features we are already presented with, namely, the state names from where each query was asked, the season and query type Based on this information we can derive the statistics as mentioned in the following tables. We got the statistics related to the number of queries per state3.7, the number of queries per crop Table3.3 as well as crop type, the number of queries per season Table3.1 and the distribution of queries based on sectors (Fig)reffig:sector as well as query types Table3.4

The data analysis gives a good picture of the agricultural landscape of India regarding which crops are popular in which state, what kind of queries are most commonly asked, and the different sectors the queries are related. For instance, the maximum number of queries asked were related to cereals, specially paddy. Also, the maximum number of queries were asked from the state of Uttar Pradesh. All of these statistics turned out to be factually accurate.

The number of queries asked during each season is shown in Table3.1. This distribution indicates that the number of farmers grows crops during Kharif season is most as compared to other seasons and hence the number of queries for this season are 54%. These analysis shows that only a limited number of unique queries are encountered while most of the queries are redundant. It also shows

that the number of queries varies drastically from state to state. The answers to each query also differs on the basis of state and district from where the query has been asked.

The Figure3.6 shows the clear view of Number of Queries asked as per Sector. By this we can understand Agricultural sector need more information to clarify farmer query.

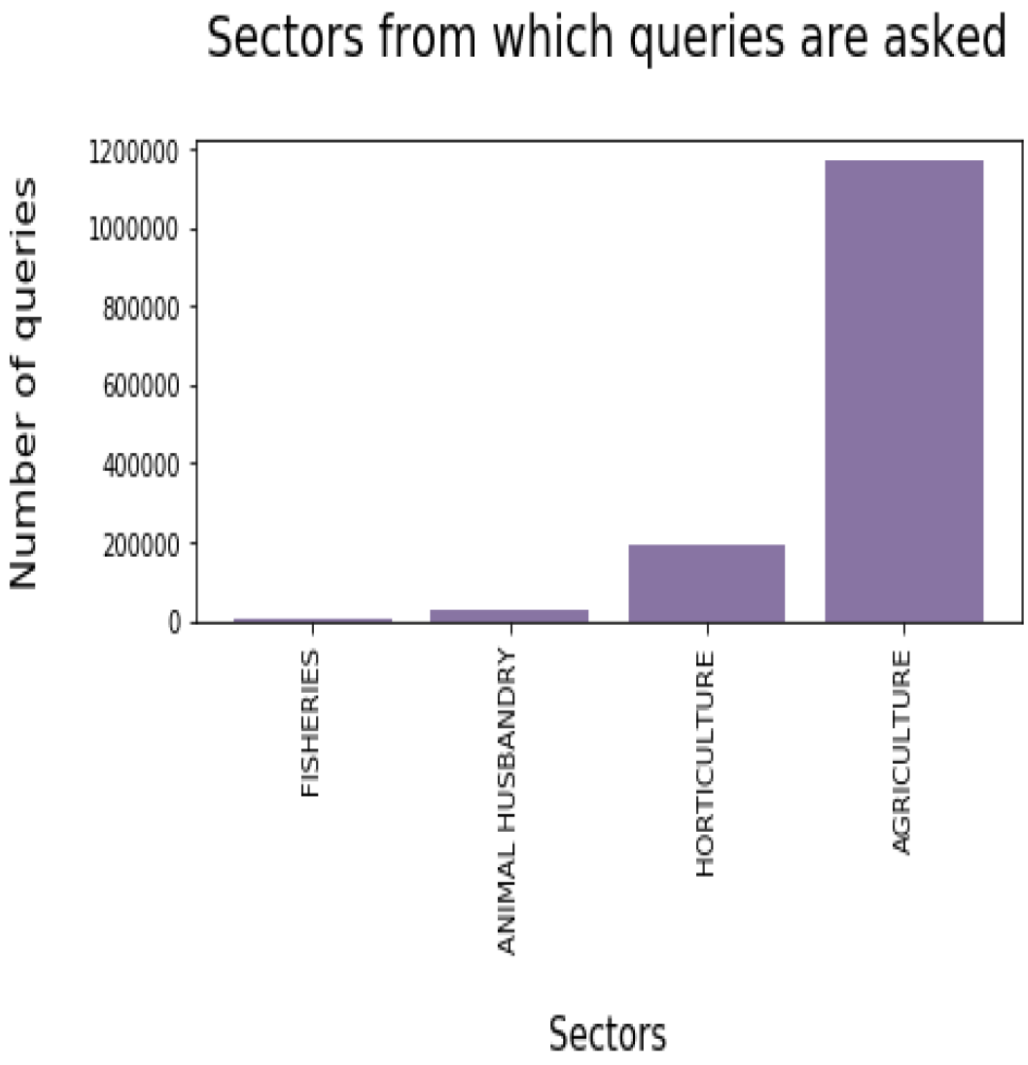

Figure 3.6: Sector Wise Details

The Figure 3.7 shows Query asked by states wise from minimum to maximum. In this analysis Uttar pradesh farmers asked huge number of queries.

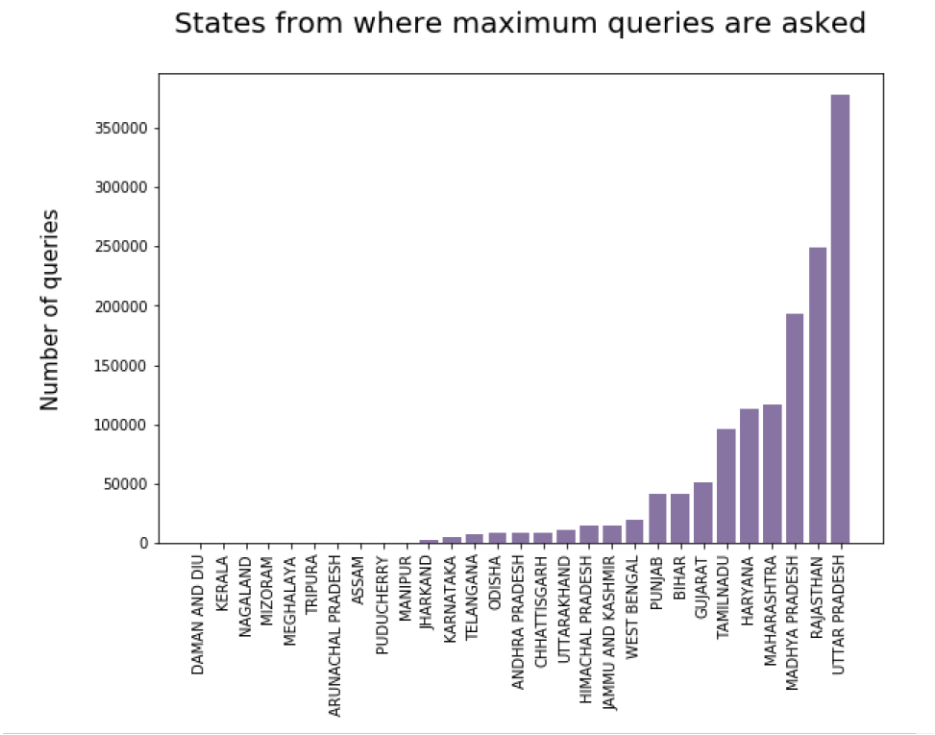

Figure 3.7: State Wise Query Received

In India there are mainly three Seasons was considered. Below table 3.1 shows the Percentage of Query's asked on each season.

SEASON	[PERCENTAGE
Karif	56.4
Rabi	28.6
Jayad	15.1

Table 3.1: DISTRIBUTION OF QUERY PAIRS AMONG SEASONS

This Data Set contains a column titled Crop Type. In which the number of types of crops can be seen, as well as how a particular species can be identified as part of a particular crop type. The percentage of queries asked in each Crop type can also be viewed in the below table3.2.

CROP TYPE	PERCENTAGE
cereals	32.5
vegetables	17.4
pulses	11.8
fruit	9.1
oil seed	8.2
fiber crop	7.3
millets	6.2
animals	7.5

Table 3.2: QUERY DISTRIBUTION BASED ON CROP TYPE

In this Query Clarifier system Crop names have a huge importance. Based on the crop name only answer can be given. The table 3.3 gives the view about percentage of query asked based on crop names.

QUERY(Crop Name)	PERCENTAGE
paddy	30.5
wheat	20.7
cotton	9.2
perl millet	7.5
sugarcane	6.3
bovine	5.8
groundnut	5.5
black gram	5.0
bengal gram	4.8
green gram	4.7

Table 3.3: QUERY DISTRIBUTION BASED OF CROP NAMES

The Query Type feature is mainly focus on what exactly the query about. Based on the Query there are several features of Query Type was defined. Below tabel3.4 shows how many percentage of query asked on each Query Type

QUERY TYPE	PERCENTAGE
weather	64.4
plant protection	17.8
government schemes	4.5
nutrition management	4.1
cultural practices	3.6
fertilizer use	2.9
variety	2.8

Table 3.4: DISTRIBUTION OF QUERY BASED ON QUERY TYPE

CHAPTER 4

IMPLEMENTATION AND RESULTS

4.1 INTRODUCTION

Implementation and Results consists of the details about the hardware and software requirements to the project and the implementations that have been performed along with their outcomes.

4.2 HARDWARE REQUIREMENTS

- **Operating System:** Windows
- **RAM :**8GB RAM or Above

4.3 SOFTWARE REQUIREMENTS

- **Coding Language :** Python, HTML, CSS
- **Framework :** Flask
- **Tool :** Google Colab , Pycharm.

4.4 LIBRARY PACKAGES

PIP: It is the package management system used to install and manage software packages written in python

TensorFlow: A Python library for speedy numerical computation created and deployed by Google. It is a platform library that can be used to create Deep Learning model directly or by using wrapping libraries that can very much simplify the process built on top of TensorFlow.

Numpy: It is an open-source extended module for Python, which provides fast pre-compiled functions on mathematical and numerical functions. NumPy enables the programme language Python with strong data structure for efficient computation of multi-dimensional arrays and matrices. The module supply a library of high-level mathematical functions to operate on these arrays.

NLTK: The Natural Language Toolkit (NLTK) is a platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP). It contains text processing libraries packages tokenization, parsing, classification.

Flask: Flask is a small and lightweight Python web framework that provides useful tools and features that make creating web applications in Python easier. Flask is very much a ”do it yourself” web framework.

Google Colaboratory: Google colab is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud. Colaboratory execute code in both python and R, save and share the analyses, and access powerful computing resources, all for free from computer browser. It is developed by google and it produces GPU support for the training on keras model.

4.5 CREATING A DATASET

Dataset is generated by downloading the data from data.gov.in database. And to make the data in a proper format with field and value context.

Step 1: Start

Step 2: Create an Automated program for downloading dataset from data.gov.in

Step 3: Initialize the value for distCode, stateCode,month, year,num.

Step 4: Get the JSON data as per the initialized values.

Step 5: If data not available increase the values and follow step 3.

Step 6: Save the data in JSON format.

Step 7: Convert the JSON file to CSV.

Step 8: End.

The Figure4.1 explains the collecting daataset from the Government data base[6]

```
...  Length of JSON:  230528
     dist=01
     states=01
     months=01
     year=2020
     data set=1
     *************
     Length of JSON:  203296
     dist=02
     states=01
     months=01
     year=2020
     data set=2
     *************
     Length of JSON:  227123
     dist=03
     states=01
     months=01
     year=2020
     data set=3
     *************
     Length of JSON:  333468
     dist=04
     states=01
     months=01
     year=2020
     data set=4
     *************
     Length of JSON:  329953
     dist=05
     states=01
     months=01
     year=2020
     data set=5
     *************
```

Figure 4.1: Raw JSON Dataset

INPUT:

Figure4.2 is the raw JSON text collected from the above step.

{"Season":"NA","Sector":"HORTICULTURE","Category":"Fruits","Crop":"Jack Fruit","QueryType":"\tPlant Protection\t",
"QueryText":"RHIZOPUS ROT MANAGEMENT IN JACK FRUIT?","KccAns":"--RECOMMENDED TO SPRAY CARBENDAZIM (BAVISTIN) 200 GRAMS/
200 LITRES OF WATER /ACRE\nకార్బెండజిం 200 గ్రాములు /200 లీటర్ల నీటికి చొప్పున కలిపి ఒక ఎకరాకు పిచికారి చేయాలి \n\n(OR)\nRECOMMENDED TO SPRAY MANCOZEB
(M-45) 600 GRAMS / 200 LITRES OF WATER / ACRE \nకలిమాంకోజెబ్ 600 గ్రాములు /200 లీటర్ల నీటికి చొప్పున \n","StateName":"PUDUCHERRY",
"DistrictName":"YANAM","BlockName":"YANAM","CreatedOn":"2021-02-05T14:53:37.027"}

Figure 4.2: Collected Sample JSON

OUTPUT:

The Collected data set was converted to CSV data file. And stored as per year wise. Figure 4.3&4.4 shows the data in CSV format.

	A	B	C	D	E	F
1	Season	Sector	Category	Crop	QueryType	QueryText
2	NA	HORTICU	0	Teak	Agriculture Mechaniza	ASKED ABOUT THE GROWING
3	NA	AGRICUI	0	Black Gra	Agriculture Mechaniza	ASKED ABOUT THE SEASON
4	JAYAD	AGRICUI	0	1137	5	VERITIES FOR RABI SEASION IN RICE
5	NA	AGRICUI	0	1075	5	dhm 117, 107 103
6	NA	AGRICUI	0	1137	Agriculture Mechaniza	ASKED ABOUT THE ZN EFFECT
7	NA	AGRICUI	0	1137	Fertilizer Use and Ava	FERTILIZER DOSAGE
8	NA	HORTICU	0	1280	Agriculture Mechaniza	ASKED ABOUT THE MANAGEMENT
9	NA	ANIMAL	0	Others	38	ASKED ABOUT THE SHEDS
10	NA	AGRICUI	0	Black Gra	2	virus disease
11	JAYAD	HORTICU	0	1279	76	ASKED ABOUT THE CONTROL OF FRUIT
12	JAYAD	AGRICUI	0	1137	3	CONTROL OF GALL MIDGE IN RICE
13	NA	AGRICUI	0	1137	Agriculture Mechaniza	asked for the controll of rice blast
14	JAYAD	AGRICUI	0	Black Gra	3	CONTROL OF THE SUCKING PEST IN BL
15	NA	AGRICUI	0	1137	Agriculture Mechaniza	rice transplanter
16	NA	AGRICUI	0	Green Gra	Agriculture Mechaniza	ASKED ABOUT THE SEASON
17	NA	HORTICU	0	1282	Agriculture Mechaniza	ASKED ABOUT THE VARITES
18	JAYAD	HORTICU	0	1280	76	ASKED ABOUT THE CONTROL OF THRIP
19	NA	AGRICUI	0	Green Gra	Agriculture Mechaniza	asked for the controll of shedding of flc
20	JAYAD	HORTICU	0	1268	99	ASKED ABOUT THE CONTROL OF FLOW
21	JAYAD	AGRICUI	0	1137	2	ASKED ABOUT THE CONTROL OF ZINC I
22	NA	AGRICUI	0	Green Gra	Agriculture Mechaniza	asked for the controll of shedding of flc
23	JAYAD	AGRICUI	0	Sunnhem	3	control of early shoot borer
24	NA	AGRICUI	0	1137	Agriculture Mechaniza	ASKED ABOUT THE ZN EFFECT
25	NA	AGRICUI	0	1137	Fertilizer Use and Ava	ZINC AND IRON DEFECIENCY
26	NA	AGRICUI	0	1075	Agriculture Mechaniza	ASKED ABOUT THE VARITES
27	NA	AGRICUI	0	1137	Agriculture Mechaniza	SEEDLINGS ZN DEFECIENCY
28	JAYAD	AGRICUI	0	Saffron	5	VARIETIES OF TURMARIC
29	JAYAD	HORTICU	0	1280	76	CONTROL OF THRIPS IN CHILLI

Figure 4.3: Conversion of JSON to CSV

F	G	H	I	J	K	L
QueryText	KccAns	StateName	DistrictNa	BlockNam	CreatedOn	
ASKED ABOUT THE GROWING	GIVEN INFORMATION AS PER THI	ANDHRA PRAD	SRIKAKULA	0	2011-01-05T08:00:00	
ASKED ABOUT THE SEASON	GIVEN INFORMATION AS PER THI	ANDHRA PRAD	SRIKAKULA	0	2011-01-05T08:00:00	
VERITIES FOR RABI SEASION IN RICE	Person 1010, a person 1001	ANDHRA PRAD	SRIKAKULA	0	2011-01-06T08:28:00	
dhm 117, 107 103	DHM 117, 107 103	ANDHRA PRAD	SRIKAKULA	0	2011-01-08T15:16:00	
ASKED ABOUT THE ZN EFFECT	SPRAY ZNSO4 2GM/1LIT OF WATI	ANDHRA PRAD	SRIKAKULA	0	2011-01-09T08:45:00	
FERTILIZER DOSAGE	RECOMMENDED INFORMATION	ANDHRA PRAD	SRIKAKULA	0	2011-01-09T10:47:00	
ASKED ABOUT THE MANAGEMENT	GIVEN INFPRMATION AS PER THE	ANDHRA PRAD	SRIKAKULA	0	2011-01-11T09:19:00	
ASKED ABOUT THE SHEDS	GIVEN INFPRMATION AS PER THE	ANDHRA PRAD	SRIKAKULA	0	2011-01-11T10:15:00	
virus disease	spraying of dimethoate 2.0 ml pe	ANDHRA PRAD	SRIKAKULA	0	2011-01-12T14:28:00	
ASKED ABOUT THE CONTROL OF FRUIT	RECOMMENDED TO SPRAY CHLO	ANDHRA PRAD	SRIKAKULA	0	2011-01-20T07:01:00	
CONTROL OF GALL MIDGE IN RICE	APPLYING CARBOFURAN 3G GRA	ANDHRA PRAD	VIZIANAGA	0	2011-01-01T08:31:00	
asked for the controll of rice blast	recommended to spray tricyclazo	ANDHRA PRAD	VIZIANAGA	0	2011-01-02T10:20:00	
CONTROL OF THE SUCKING PEST IN BL	SPRAYING CHLOROPYRIPHOS 2m	ANDHRA PRAD	VIZIANAGA	0	2011-01-05T15:26:00	
rice transplanter	required information has given	ANDHRA PRAD	VIZIANAGA	0	2011-01-08T15:16:00	
ASKED ABOUT THE SEASON	GIVEN INFORMATION AS PER THI	ANDHRA PRAD	VIZIANAGA	0	2011-01-12T14:03:00	
ASKED ABOUT THE VARITES	Soo the finest father, Namqa	ANDHRA PRAD	VIZIANAGA	0	2011-01-13T07:56:00	
ASKED ABOUT THE CONTROL OF THRIP	RECOMMENDED TO SPRAY CARB	ANDHRA PRAD	VIZIANAGA	0	2011-01-13T15:44:00	
asked for the controll of shedding of flc	recommended to spray planofix 2	ANDHRA PRAD	VIZIANAGA	0	2011-01-15T08:05:00	
ASKED ABOUT THE CONTROL OF FLOW	RECOMMENDED TO SPRAY PLAN	ANDHRA PRAD	VIZIANAGA	0	2011-01-22T13:54:00	
ASKED ABOUT THE CONTROL OF ZINC I	RECOMMENDED TO SPRAY ZZINC	ANDHRA PRAD	VISAKHAPA	0	2011-01-08T07:52:00	
asked for the controll of shedding of flc	recommended to spray planofix 2	ANDHRA PRAD	VISAKHAPA	0	2011-01-15T08:27:00	
control of early shoot borer	applying sevidal 74kg/1 acer	ANDHRA PRAD	VISAKHAPA	0	2011-01-23T14:04:00	
ASKED ABOUT THE ZN EFFECT	SPRAY ZNSO4 2GM/1LIT OF WATI	ANDHRA PRAD	VISAKHAPA	0	2011-01-24T10:05:00	
ZINC AND IRON DEFECIENCY	SPRAYING OF ZINC SULPHATE AN	ANDHRA PRAD	VISAKHAPA	0	2011-01-26T19:28:00	
ASKED ABOUT THE VARITES	Sow DHM-113	ANDHRA PRAD	VISAKHAPA	0	2011-01-28T12:06:00	
SEEDLINGS ZN DEFECIENCY	SPRAY THE 2 G ZINC SULPHATE PI	ANDHRA PRAD	EAST GOD	0	2011-01-02T15:09:00	
VARIETIES OF TURMARIC	Maidhukuru, Takiipeta, C.L.L.L.-3	ANDHRA PRAD	EAST GOD	0	2011-01-04T06:51:00	
CONTROL OF THRIPS IN CHILLI	SPRAYING KARBALRIL 3gr/1 lit W	ANDHRA PRAD	EAST GOD	0	2011-01-06T10:01:00	

Figure 4.4: Conversion of JSON to CSV

4.6 DATA PRE-PROCESSING

Data Cleaning consists of several Pre-processing step like Punctuation, Language Translation, Stop word Removal, Stemming. These task can also be performed by in-built libraries.

Step 1: Start

Step 2: Get the collected data in CSV format

Step 3: Assign the CSV data to df

Step 4: Initialize the Pre-Processing step.

Step 5: Punctuation, Language Translation, Tokenization.

Step 6: Stop word Removal,Stemming.

Step 7: Save the Pre-Processed dataset in CSV file

Step 8: End.

INPUT:

Input for this module are CSV file.Figure 4.34.4 shows the data in CSV format.

OUTPUT:

	A	B	C	D	E	F
1	Query	QueryType	StateName	DistrictName	CreatedOn	Answer
2	RED PALM WEEV	Plant Prote	TAMILNADU	TIRUPUR	20160118T21	RECOMMEND ROOT FEED
3	RED PALM WEEV	Plant Prote	TAMILNADU	NAGAPATTINAM	20160116T11	RECOMMEND ROOT FEED
4	RED PALM WEEV	Plant Prote	TAMILNADU	THANJAVUR	20160103T07	RECOMMEND ROOT FEED
5	RED PALM WEEV	Plant Prote	TAMILNADU	ERODE	20170127T10	RECOMMEND APPLIC GRE
6	RED POD BORER	Plant Prote	ANDHRA PR	KURNOOL	20160117T06	RECOMMEND SPRAY FLUB
7	RED PUMKIN BE	Plant Prote	UTTAR PRA	MIRZAPUR	20160126T19	SPRAY CLOROSAYP 400 ML
8	RED PUMPKIN BI	Plant Prote	UTTAR PRA	MIRZAPUR	20170215T20	DIMETHO 30 % EC 250 ML
9	RED PUMPKIN BI	Plant Prote	UTTAR PRA	BADAUN	20170323T10	DIMETHO 30 EC 500 WA D
10	RED PUMPKIN BI	Plant Prote	UTTAR PRA	GORAKHPUR	20170308T10	INFORM CONTACT BLOCK
11	RED PUMPKIN BI	Plant Prote	UTTAR PRA	ETAWAH	20170324T20	TARBOOJ KE PAUDH PAR M
12	RED PUMPKIN BI	Plant Prote	ODISHA	MAYURBHANJ	20160108T18	RECOMMEND SPRAY QUIN
13	RED PUMPKIN BI	BioPesticid	ODISHA	BALASORE	20170208T08	SPRAY MALATHION 2 ML L
14	RED PUMPKIN BI	Plant Prote	ODISHA	MAYURBHANJ	20160108T18	RECOMMEND SPRAY QUIN
15	WHEAT URA SPR	Fertilizer U	UTTAR PRA	SHAHJAHANPUR	20160115T15	UREA 3 KILO ACR SPRAY K
16	WHEAT UREA AC	Field Prepa	UTTAR PRA	AZAMGARH	20170105T19	40 KG UREAACR PRYOG KR
17	WHEAT UREA AL	Fertilizer U	UTTAR PRA	PILIBHIT	20160111T15	SPRAY UREA 8KG.ACRE+ZI
18	WHEAT UREA DA	Fertilizer U	UTTAR PRA	KHERI	20160108T18	APPLI UREA 50 KG + ZYEM
19	WHEAT UREA DA	Fertilizer U	UTTAR PRA	SAHARANPUR	20160104T14	APPLI UREA50 KG + ZYEM1
20	WHEAT UREA DA	Plant Prote	UTTAR PRA	BAREILLY	20160116T14	APPLI UREA 50 KG ACR
21	WHEAT UREA DA	Weather	UTTAR PRA	SHAHJAHANPUR	20160107T11	UREA 50 KG ACR PRIYOG K
22	WHEAT UREA KA	Fertilizer U	UTTAR PRA	ETAH	20160122T11	50 KG UREA ACR PRYOG K
23	WHEAT UREA KA	Fertilizer U	UTTAR PRA	SAHARANPUR	20170104T13	2 KG UREA100 LITER WATE
24	WHEAT UREA KA	Fertilizer U	UTTAR PRA	FAIZABAD	20170208T14	SPRAY UREA 2 % WHEAT C
25	WHEAT UREA KA	Fertilizer U	UTTAR PRA	GORAKHPUR	20170202T17	SPRAY UREA 2 %
26	WHEAT UREA KA	Fertilizer U	UTTAR PRA	GONDA	20170213T17	SPRAY UREA 2 % WHEAT C
27	WHEAT UREA KA	Fertilizer U	UTTAR PRA	GONDA	20170205T20	SPRAY UREA 2 % WHEAT C
28	WHEAT UREA KA	Fertilizer U	UTTAR PRA	GONDA	20160111T18	SPRAY UREA 2 % SALUT

Figure 4.5: Pre-Processed Data Set

4.7 DESIGNING DATA FRAME

By the preprocessed data set we can create a data frame which contain Query, Query type, State, District, Time of Query and List of Answers for the particular Query. The data frame is made in the form of dictionary Query is the Key value and all other data are values for the key in the form of list.This Data frame is more useful for developing the model.

Step 1: Start
Step 2: Get the collected data in CSV format after Pre-Processing.
Step 3: Create an empty Dictionary.
Step 4: Assign the key as an unique query.
Step 5: Add the value for every unique keys.
Step 6: Value for the keys contains Query type, State, District, Time of Query and List of Answers.
Step 7: Append the value one another for every unique query.
Step 8: End.

INPUT :

Input for this module are CSV file.Figure 4.5 shows the data in CSV format for easy understanding.

OUTPUT:


```
"FRUIT CRACK DUM STICK": [["Plant Protection", "MAHARASHTRA", "AHMADNAGAR", "20180212T10:48:32.077", "SPRA
"FRUIT CRACK GRAPE": [["Fertilizer Use and Availability", "MAHARASHTRA", "SANGLI", "20170109T12:04:36.2", "
"FRUIT CRACK GRAPES": [["Plant Protection", "MAHARASHTRA", "SOLAPUR", "20170130T07:55:01.273", "SPRAY CALS
"FRUIT CRACK GUAVA": [["Nutrient Management", "BIHAR", "LAKHISARIA", "20190117T08:54:56.82", "SPRAY BORON
"FRUIT CRACK INFORM BANANA": [["Cultural Practices", "UTTAR PRADESH", "ALLAHABAD", "20180119T18:45:37.707",
"FRUIT CRACK JACK FRUIT": [["Plant Protection", "JHARKAND", "HAZARIBAGH", "20160301T19:55:01.25", "SPRAY B(
        ["Nutrient Management", "BIHAR", "PURNEA", "20180327T15:00:45.13", "SPRAY BORON 2 GRAM PER LITR WA
        ["Fertilizer Use and Availability", "WEST BENGAL", "BIRBHUM", "20160203T16:19:53.397", "RECOMMEND
"FRUIT CRACK KNOLKHOL": [["BioPesticides and BioFertilizers", "ODISHA", "GANJAM", "20180119T11:46:26.153",
"FRUIT CRACK LEMON": [["Plant Protection", "ODISHA", "KALAHANDI", "20170108T18:13:56.51", "SPRAY BORAX 4 GM
        ["Nutrient Management", "BIHAR", "MADHEPURA", "20180205T11:13:14.12", "SPRAY BOREX  2GMLIT WATER"],
        ["Plant Protection", "JHARKAND", "PALAMU", "20160122T18:50:25.083", "SPRAY BORAX  2GMLIT WATER"],
        ["Plant Protection", "ODISHA", "KALAHANDI", "20170109T18:48:08.043", "APPLI BORAX CRACK LEMON"]],
"FRUIT CRACK LITCHI": [["Plant Protection", "BIHAR", "PURBA CHAMPARAN", "20160511T17:15:48.11", "APPLI BORA
        ["Plant Protection", "BIHAR", "BHOJPUR", "20160511T20:58:46.957", "SPRAY BOREX 2GRAMLIT WATER LITCH
        ["Plant Protection", "BIHAR", "BHAGALPUR", "20160430T14:48:00.757", "FRUIT CRACK LITCHI  SPRAY BORI
        ["Plant Protection", "BIHAR", "PATNA", "20160514T16:08:34.137", "SPRAY BORAX  2GMLIT WATER"],
        ["Plant Protection", "ODISHA", "BALASORE", "20160119T07:52:18.863", "FRUIT CRACK LITCHI SPRAY ZINC
        ["Plant Protection", "BIHAR", "KHAGARIA", "20160429T20:47:13.417", "BORON 10GRAMPLANT"]],
"FRUIT CRACK MANAG TOMATO": [["Nutrient Management", "TAMILNADU", "PUDUKKOTTAI", "20170123T14:43:37.07", "F
"FRUIT CRACK MANGO": [["Plant Protection", "BIHAR", "SAMASTIPUR", "20160425T21:12:23.57", "       2
        ["Plant Protection", "BIHAR", "AURANGABAD", "20160506T18:56:02.99", "SPRAY BOREX 2GM PER LIT WATER'
        ["Plant Protection", "BIHAR", "BHOJPUR", "20160202T08:40:46.4", "FRUIT CRACK MANGO SPRAY BOREX  2GM
        ["Nutrient Management", "BIHAR", "JAMUI", "20180319T13:04:35.723", "APPLI BORON ROOT  2 GM PER LITR
        ["Plant Protection", "BIHAR", "PATNA", "20160218T15:17:40.49", "FRUIT CRACKING SPRAY BOREX  2GMLIT
        ["Plant Protection", "BIHAR", "NALANDA", "20160202T07:05:38.03", "FRUIT CRACK MANGO SPRAY BOREX  2(
        ["Plant Protection", "JHARKAND", "KHUNTI", "20160313T06:55:34.457", "SPRAY BORAX  2GMLIT WATER"],
        ["Weather", "BIHAR", "PATNA", "20160509T19:40:32.247", "FRUIT CRACK MANGO SPRAY BOREX  2MLLIT WATER
        ["Plant Protection", "BIHAR", "PATNA", "20160509T18:50:14.593", "FRUIT CRACK MANGO SPRAY BOREX  2ML
        ["Plant Protection", "BIHAR", "PATNA", "20160501T20:37:28.047", "FRUIT CRACK MANGO SPRAY BOREX  2ML
        ["Nutrient Management", "BIHAR", "PURNEA", "20180311T13:22:15.473", "SPRAY BORON  2 GM PER LITR WA
        ["Plant Protection", "BIHAR", "ROHTAS", "20160514T17:19:50.153", "SPRAY BOREX  2MLLIT WATER"],
        ["Plant Protection", "BIHAR", "ROHTAS", "20160509T19:55:06.58", "FRUIT CRACK MANGO SPRAY BOREX  2ML
        ["Plant Protection", "BIHAR", "PURBA CHAMPARAN", "20160506T15:55:37.697", "FRUIT CRACK MANGO SPRAY
        ["Plant Protection", "BIHAR", "ROHTAS", "20160508T20:26:55.693", "SPRAY BOREX  2MLLIT WATER"],
        ["Plant Protection", "BIHAR", "PASHCHIM CHAMPARAN", "20160416T18:50:26.37", "APPLI BORON 5GRAMPLAN
```


Figure 4.6: Sample Data Frame Creation

4.8 DEVELOPING MODELS

For Model training we use Sentence Embedding technique which converts the whole sentence in a vector form. The trained model consist a matrix in 768 dimension. For Sentence embedding bert-base-nli-mean-tokens is more

suitable model for this NLP project.

Step 1: Start

Step 2: Get the collected data in CSV format after Data-Frame Creation.

Step 3: Create an empty list

Step 4: Append all the keys value from Data Frame into the list.

Step 5: Use Bert-base-nli-means-token model for sentence embedding.

Step 6: Use the created list for training the model

Step 7: Save the trained model

Step 8: End.

INPUT:

```
ADDRESS KVK SIVASAGAR
ADDRESS NABARD BANK AHMEDNAGAR
ADDRESS NABARD BANK DHULE
ADDRESS NABARD BANK JALNA
ADDRESS NABARD BANK NASIK
ADDRESS NABARD BANK YAVATM
ADDRESS NASIK KVK
ADDRESS NATION CENTER ORGAN FARM
ADDRESS NHRDF CENTR
ADDRESS PERAMBALUR AGRICULTUR ENGIN DEPART
ADDRESS PERAMBALUR VETERINARI UNIVERS TRAIN RESEARCH CENTR
ADDRESS PHONE RATHINDRA KVK
ADDRESS PROBLEM REGISTR
ADDRESS RESEARCH STATION HYDERABAD
ADDRESS SHETKARI MAGZINS
ADDRESS SHETKARI MASHIK PUNE
ADDRESS SOIL TEST LAB
ADDRESS SOIL TEST LABORATORI SOLAPUR
ADDRESS SOIL TESTING
ADDRESS SONARPUR KVK
ADDRESS THANJAVUR SOIL TEST LABORATORI
ADDRESS TIRUVANNAMALAI VETERINARI UNIVERS TRAIN RESEARCH CENTR
ADDRESS UNIVERS AGRICULTUR SCIENCES DHARWAD UAS
ADDRESS VETERNARI HOSPIT
ADDRESS WHEAT POLISH
ADDRESSAND CONTACT NUMBER KRUSHI VIGYAN KENDRA
ADDTIT APPLIC RECOMMEND NPK ADD 1 KG GYPSUM 50 G BORAN 5 KG NEEM OIL CAKEPALM
ADDU BONA CHAHT HAI KAB BOYE
ADE ARRIVALS UNITS MIN RS MAX RS MODAL RS BELTHANGADI COPRA AVERAGE 3 QUINTAL
ADE ARRIVALS UNITS MIN RS MAX RS MODAL RS BENGALURU LOCAL AVERAGE 6520 QUINTAL
ADEU KE PADE PER MAHU KA PARKOP
ADEXAR DESCRIPT
ADH CONTACT NUMBER RAJAMPETA
ADH HORICULTUR NELLOR
```

Figure 4.7: List of Unique Query's

OUTPUT:

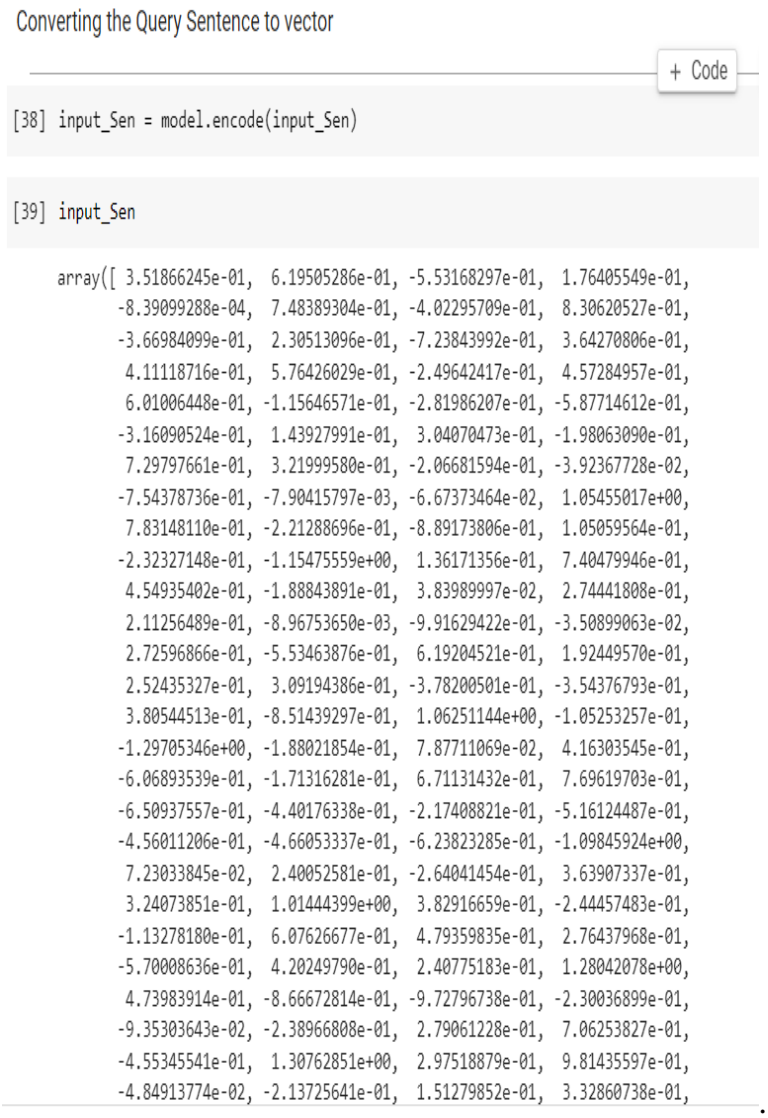

Figure 4.8: Vectorized Query

4.9 QUESTION-ANSWER MAPPING

In this answer mapping we use Cosine similarity to find the similarity of vector from the database and Lesk Algorithm for Ranking the list of answer got from similarity check.

Step 1: Start

Step 2: Get the Input query form User

Step 3: Follow the steps in Data Cleaning module

Step 4: Convert the Input query to Vectorized matrix using Bert model.

Step 5: Get the saved model from model training module

Step 6: Compare the matrix of user query in trained model.

Step 7: Get the predicted Query and list of answers

Step 8: Use Answer Ranking method[10] to find suitable answer.

INPUT :

Input for this module in Vectorized form which contain 768 dimension matrix format. Figure4.8 gives the clear view of matrix.

OUTPUT:

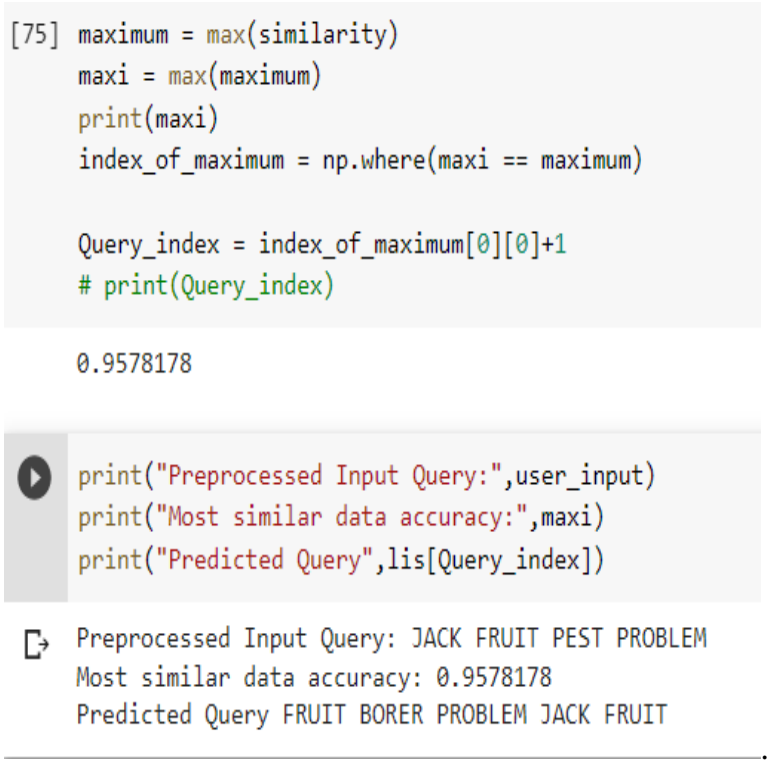

Figure 4.9: Predicted Query

4.10 RESULTS AND PERFORMANCE ANALYSIS

This chapter should provide the details of results and the analysis of your work. Depending on the type of project, there may not be analysis. In such cases, mention the title as “Results” or “Testing and Results”.

4.10.1 PERFORMANCE ANALYSIS

For the Performance analysis, we wanted to capture similarity between the input sentence to the model and output predicted sentence from the model and use this to determine whether the two sentences are same

We found that none of the standard scientific metrics to be suitable for evaluating our model. Because of the improper and inconsistent structure of question-answer pairs regarding language usage, we had to design a metric from scratch. Taking inspiration from Jaccard and Lesk similarity[10] metrics, we devised two metrics - modified Jaccard and modified Lesk scores in order to evaluate our model.

Our metric can be thought of as the amount of similarity between two sentences - input and prediction. Thus being able to find this value should give us a direct understanding of how our model is performing.

A.MODIFIED JACCARD SCORE

We define our modified Jaccard score as the number of words in the intersection of the given question (known sentence) and the predicted question (predicted sentence).We add 1 to the denominator to avoid zero division.

$$Jaccard = \frac{count(KnownSent \cap PredictedSent)}{count(KnownSent) + 1} \tag{4.1}$$

In this method, we simply use the words in the sentences as our parameters.

A.MODIFIED L LEST SCORE We first used words from meanings for various senses of words to create a gloss bag of words. We define our metric as the number of common words in the gloss bag of input question (known sentence) and the predicted question (predicted sentence) divided by the number of words in the gloss bag of the input question (known sentence).

$$Lesk = \frac{count(gloss(knownSent) \cap gloss(PredictedSent))}{count(gloss(knownSent) + 1} \tag{4.2}$$

For this equation also e add 1 to the denominator to avoid zero division.

4.10.2 EVALUATION

In order to evaluate our metric, we labeled some test data queries and calculated our modified Jaccard scores and modified Lesk scores for the prediction of the test data questions. Using these predictions and the ground truth we then define a threshold for both scores. The threshold tells the model which predictions are to be considered as good results. We accordingly use the metrics for ranking our answers, where the final predicted answer is given by:

$$outputanswer = argmax[score(question, answer)] \tag{4.3}$$

4.10.3 RESULTS

Using the Modified Lesk score metric, our model was able to obtain an accuracy of about 65% without synonym elimination and entity extraction. One key observation was that the crop names were important determiner while comparing the most similar queries. We thus performed entity extraction for the

crop names. We had observed that the accuracy jumped from 65% to 86% after using entity extraction. We then varied the dimension to improve accuracy. As demonstrated by the Fig. 4, the best performance of the model was observed at 768 number of dimensions for the embedding.

Top N	Jaccard	Modified Lesk
Top 1	64%	86%
Top 2	59%	89%
Top 3	70%	91%

Table 4.1: METRIC SCORE COMPARISON IN TOP-N MOST SIMILAR OUTPUT QUERIES

4.10.4 EXPERIMENTAL RESULTS OF THE APPLICATION

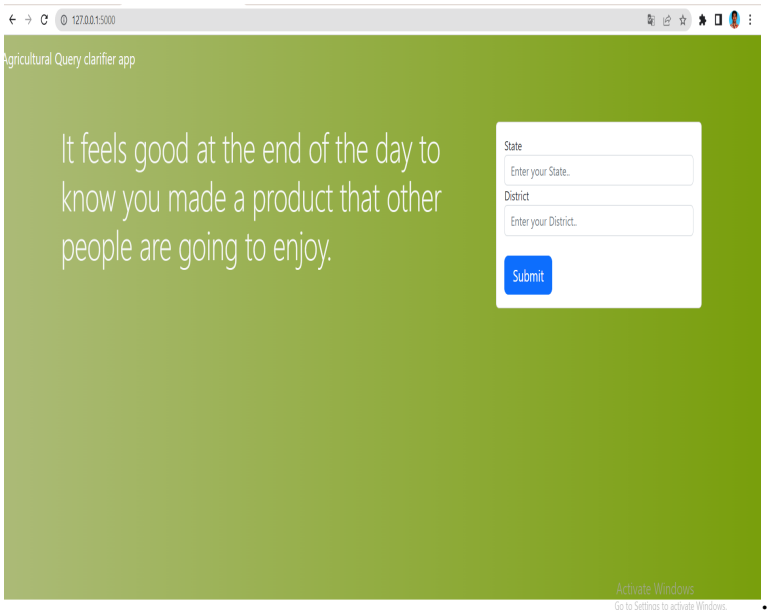

Figure 4.10: User Interface First page

In this Section , figures4.10 ,will represent the User interface of the user enter their Location here. This entered location will be used for the Answer ranking system[10].

After the user entered the location they will redirect to the Main page of our application. Here user can enter their query and get an appropriate predicted answer for the same.

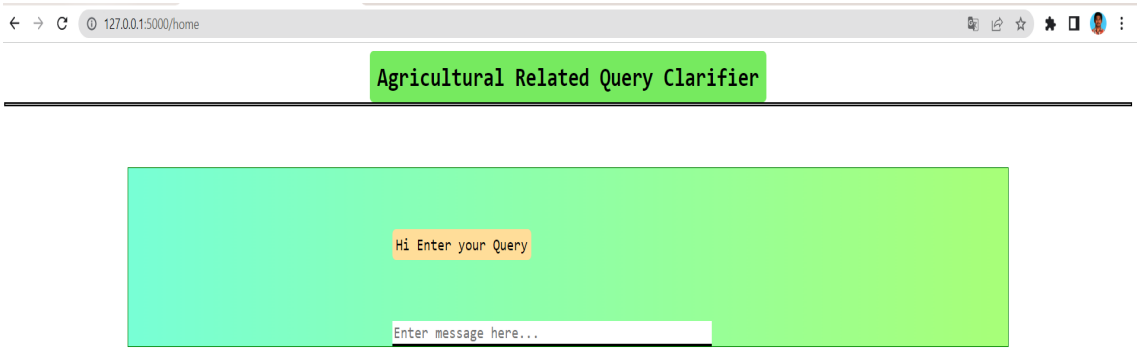

Figure 4.11: User Interface Main Page

The below figure 4.12 gives the result for the user query in terminal view.

```
Query input: tips for mango tree pest control
Predicted input: MANGO FRUIT PEST CONTROL
Answer: APPLI COPPER OXYCHLORID [ 50WP ]  3GMLITWATERTRIAZOPHOS [ 40EC ]  2MLLITWATER
```

Figure 4.12: Answer for the given query Terminal view

The below figure 4.13 gives the result for the user query in the User Interface

Figure 4.13: Answer for the given query using User Interface

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

This Query clarifier system can positively impact more in under served communities by solving queries related to agriculture, horticulture and animal husbandry using natural language technology. The farmer will be able to receive agricultural information as well as localized information such as the current market prices of various crops in his/her district. A farmer can directly message our AI enabled the system and get an answer. This system would enable the farmer to ask any number of questions, anytime, which will in turn help in spreading the modern farming technology faster and to a higher number of farmers.

Moreover, this system found that most of the queries related to localized information such as weather and market prices were redundant. In Question Answer system can answer maximum queries on its own without any human intervention with high accuracy. This will lead to better utilization of human resource and avoid unnecessary costs in setting up new call center. Above all, we believe that the system helps in analyzing the farmers' mindset and the structure of the Agricultural Sector in India. Thus, our decision support system uses all the available resources judiciously to tackle the problem of lack of awareness and information in the agricultural sector in India.

5.2 FUTURE WORK

For the future, we plan to implement multilingual support for the Query Clarifier system with voice-over support and entity extraction from answers for generating knowledge graphs.The system also provides an option that enables the farmer to ask questions directly to the KCC employees if and when necessary

REFERENCES

[1] Telecom Regulatory Authority of India. ”Annual Report”. 19-25,2020-2021.

[2] Government of India. ”Farmers Portal”. *https://www.farmer.gov.in/*, 2013.

[3] Government of India. ”Kisan Call Center”. *https://dackkms.gov.in/account/login.aspx*, 2022.

[4] G. Ifrim M. Ramanath G. Kasneci, F. M. Suchanek and G. Weikum. ”NAGA: Searching and Ranking Knowledge”, journal = In the Proceedings of IEEE 24th International Conference on Data Engineering, year = 953-962,2008,

[5] Robin Jia Rajpurkar, Pranav and Percy Liang. ”Know What You Don’t Know: Unanswerable Questions for SQuAD”. *rXiv preprint arXiv:1806.03822*, 2018.

[6] Government of India. ”Open Government Data Platform India”. *https://data.gov.in/*, Latest Access:2022.

[7] Alexsandro Fonseca Belainine Billal and Fatiha Sadat. ”Efficient Natural Language Pre-Processing for Analyzing Large Datasets”. *In the Proceedings of IEEE International Conference on Big Data (Big Data)*, 41-56,September 2016.

[8] Nils Reimers and Iryna Gurevych. ”Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks”. *In the Proceedings of 2019 International Conference on Empirical Methods in Natural Language Processing*, 256-280, 2019.

[9] Adhistya Erna Permanasari Alfirna Rizqi Lahitani and Noor Akhmad Setiawan. ”Cosine Similarity to Determine Similarity Measure: Study Case in Online Essay Assessment”. 78-89,2016.

[10] S. Banerjee and T Pedersen. ”An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet”. *Computational Linguistics and Intelligent Text Processing, Third International Conference*, 136–145,2002.