Query Answering for Kisan Call Centerwith LDA/LSI

Sandeep Kumar Mohapatra,Coginzant Technology Solutions, Bangalore,India-560103
Email:
SandeepKumar.Mohapatra@cognizant.com

Anamika Upadhyay, Aricent Technologies Private Limited, Bangalore, India-560103
Email: anamikaupadhyay@gmail.com

Abstract –With the enhanced storage and processing capacity of the computers in today's world we have lots of unprocessed data in every domain. This data has gained interest of data scientist of all over the world to get some meaningful information from the raw data. We are here processing kisan call center data. First the preprocessing of data is done which is then converted to a similarity matrix which we save in a database using mongoDb. This prepared data will now be fed to the gensim TI-IDF model for text transformation. After this we have used another two methods of topic modelling to be used with TI-IDF model. We are concerned in getting the query answers efficiently from the data collected by usingLatent Dirichlet allocation (LDA) and Latent Semantic Indexing (LSI) in pipeline to the TI-IDF model. One algorithm which will be used with pipeline to the TI-IDF is LDA. LDA is a technique that automatically discovers topics that these documents contain. Another model which we are exploring is the LSI which also will be used with TI-IDF model for preparing the model for getting query answers efficiently.On these parameter we will be building the model training and finally will compare the models for finding which model works best in terms of efficiency, accuracy and complexity.

Keywords: TI-IDF, LDA, LSI, mongoDb, information retrieval.

I. INTRODUCTION:

India being agriculture rich country, most pf the population is dependent on agriculture for their livelihood. The farming techniques followed in India are basically traditional and they lack in technical expertise hence they requires guidance of experts. This will not only help in better yield but also help in managing their day today work. To provide such a platform to the farmers of the country government of India has launchedKisan call center(farmers help line) India on January 21 2004 to help the farmers. This is a toll free number which can be reached from any part of the country and farmers can get their query answers in their regional language. This helpline has got a huge response, there has been lakhs of calls from all over the country for getting the solutions to the difficulties they face during their day to day agricultural works.

These calls which the farmers called are being handled by experts, whose expertise is in agriculture which is the front end to the farmers. The other team being experts in informationand communication Technology. Both have their own specialized persons of their respective domains to make this framework work. The information and technology wing captured the caller's details andthe query and the answers to the queries provided by the kisan call center team. All the details are saved for further analysis for the type of problems in particular area and their resolutions provided by the experts. Government of India has given open access to the data for these details [1]. In our current work we have captured the data from this source for one year of a particular state the data includes district wise - month Wise details of queries asked by farmers and answers given by given by the kisan call center executives.

In this paper we will be using information retrieval techniques [2] to process the data collected for the entire year and will try to find out how efficiently and accurately we can get the query answers from this data. The common query answers will help the persons in agriculture sector to find out major problems which can be area-wise, crop-wise and problem wise analyses within the time-space framework and provides preventive and corrective solutions in much advance such that the productivity can be increased both qualitative and quantitative aspects. We are trying to build a framework which will accept one query text from the user and from the entire database it will fetch most relevant query asked and their respective answers given by the executives. This framework when incorporated in the system, the call time for each farmers will be reduced, also they can get their answers faster. If they missed out to listen to some answers it will be handy to then to refer and find other related problems and answers, which might be related to their problem.

This information obtained from the framework will not be useful to the farmers of the country but also can be used by the advisors of agriculture sector professionals to understand most common problems faced, and most common issues related to some particular crop and so on. This information they can use to train the farmers of an area in advance such that they can be prepared for certain unwanted calamities such that to identify pest attacks in any particular geographical area, get the relevant crop information.

To prepare the model we are using the algorithms tharare unsupervised, which means no human input is necessary – you only need a corpus of plain text documents. Once these statistical patterns are found, any plain text documents can be expressed in new, semantic representation, and queried for topical similarity against other documents. We have the dataset which is the query text is in the form of short text segments and with different contexts. The dataset has queries related to different problems which can be found in agriculture sector like pesticides, herbicides, varieties of seeds, livestock, policy issues, government interventions, price of commodity etc. We will be addressing the issues of vector formation of short texts fragments. We will be evaluating the text segment with semantic similarity task. One of the major component used for building the model is gensim python library. Gensim is scalable, robust, platform independent, open source and similarity queries. We are harnessing the similarity queries feature of gensim, which is helpful in fast indexing of documents and their semantic representation [5]. For this we are using genism's TI-IDF model in pipeline with LDA and LSI.

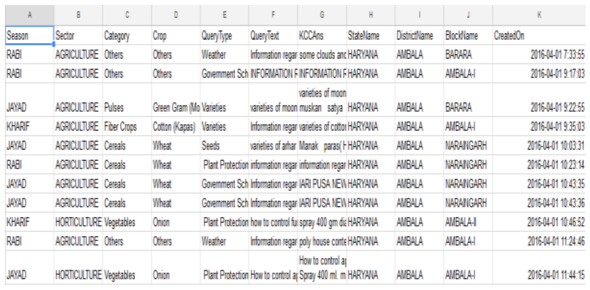
II. DATA COLLECTION AND PREPROCESSING:

Data Preparation and preprocessing is a very important step when it comes in processing and retrieving information from some raw data. This step is a compulsory step for achieving better accuracy and performance of algorithms in the Machine Learning and Deep Learning.

Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. That data may contain some unwanted and irrelevant information which might be the cause for failure of the algorithm. So to remove these discrepancies and make the data usable for the machine learning algorithms we have to follow certain steps before execution of the analysis through the algorithm. Generally data preprocessing we have following steps: [6]

- Data Cleaning – It is also known as Data cleansing is the process of finding and removing corrupt or inaccurate data records from the raw dataset. This process is to identify inaccurate and irrelevant parts in the dataset which can be replaced, modifies or deleted to make the data clean and usable
- Data Integration - this step generally collates the data from different sources for the same problem such that we get complete information for a particular problem.
- Data Transformation - this technique generally removes noise from the raw dataset. This can be done by regression and clustering.
- Data Reduction - Data Reduction implies reducing the data but without compromising integrity of it.

We have collected data for a year from the portal of government [1]. Within a year more than 40 lakhs questions has been addressed by the team. Below bar graph represents number of calls which has been handled by the kisan call centre during 2016 and 2017 [4]. This is huge dataset to build a framework to get answers to the queries more quickly. In the portal we have data for 2015, 2016 and 2017. The dataset fields and attributes which we are working on is given below. We will be basically focusing on the query text and query answers for building and testing our model. The downloaded dataset was for different months in form of csv files. These files we have merged for entire year of all the districts of Haryana. Before providing this file to the model we have prepared the data for the model, for better results. For text data we have major challenges for data preparation. Also complexity of natural language add on to the issues with text mining.The ambiguity problem is major challenge in natural language, one word in many cases may have multiple meanings also in reverse multiple words may have same meaning. This ambiguity leads to noise in extracted information, but this cannot be eliminated. To remove these discrepancy we have to preprocess the data and make the data smooth by removing any inconsistency and outliers for better performance of the prediction algorithms.



A	B	C	D	E	F	G	H	I	J	K
Season	Sector	Category	Crop	QueryType	QueryText	KCCAns	StateName	DistrictName	BlockName	CreatedOn
RABI	AGRICULTURE	Others	Others	Weather	Information regar	some clouds and	HARYANA	AMBALA	BARARA	2016-04-01 7:33:55
RABI	AGRICULTURE	Others	Others	Government Sch	INFORMATION F	INFORMATION F	HARYANA	AMBALA	AMBALA-I	2016-04-01 9:17:03
JAYAD	AGRICULTURE	Pulses	Green Gram (Mo	Varieties	varieties of moon	varieties of moon muskan satya	HARYANA	AMBALA	BARARA	2016-04-01 9:22:55
KHARIF	AGRICULTURE	Fiber Crops	Cotton (Kapas)	Varieties	Information regar	varieties of cottor	HARYANA	AMBALA	AMBALA-I	2016-04-01 9:35:03
JAYAD	AGRICULTURE	Cereals	Wheat	Seeds	varieties of arhar	Manak paras(H	HARYANA	AMBALA	NARAINGARH	2016-04-01 10:03:31
RABI	AGRICULTURE	Cereals	Wheat	Plant Protection	information regar	information regar	HARYANA	AMBALA	NARAINGARH	2016-04-01 10:23:14
JAYAD	AGRICULTURE	Cereals	Wheat	Government Sch	Information regar	IARI PUSA NEW	HARYANA	AMBALA	NARAINGARH	2016-04-01 10:43:35
JAYAD	AGRICULTURE	Cereals	Wheat	Government Sch	Information regar	IARI PUSA NEW	HARYANA	AMBALA	NARAINGARH	2016-04-01 10:43:36
KHARIF	HORTICULTURE	Vegetables	Onion	Plant Protection	how to control fu	spray 400 gm dia	HARYANA	AMBALA	AMBALA-II	2016-04-01 10:46:52
RABI	AGRICULTURE	Others	Others	Weather	Information regar	poly house conte	HARYANA	AMBALA	AMBALA-I	2016-04-01 11:24:46
JAYAD	HORTICULTURE	Vegetables	Onion	Plant Protection	How to control ap	How to control ap Spray 400 ml. m	HARYANA	AMBALA	AMBALA-I	2016-04-01 11:44:15

Figure 1 – Snippet of Kisan Call center dataset

The dataset which has been tested for our algorithm is 1 year data for the state of Haryana, India which we have collected from the government of India portal [1].For the kisan call center data we have prepared the dataset by removing the punctuations and single words. This prepared data will now be fed to the genimti-idf model for text transformation. After this we have used another two methods of topic modelling to be used with ti-idf model. One model which will be used with pipeline to the TI-IDF is Latent Dirichlet allocation (LDA). LDA is a technique that automatically discovers topics that these documents contain.

Another model which we are exploring is the LSI (Latent Semantic Analysis) which also will be used with ti-idf model for preparing the model for getting query answers efficiently.

III. SYSTEM MODEL:

One of the major component used for building the model is gensim python library. Gensim is scalable, robust, platform independent, open source and similarity queries. We are harnessing the similarity queries feature of gensim, which is helpful in fast indexing of documents and their semantic representation [5]. This TF-IDF vector model will be used by the algorithms LDA and LSI for finding the query answers.

We are using base of TI-IDF to frame the unit vector matrix [3] which will be passed to LDA and LSI for getting the query answers. We are building a model on query texts which we will be representing in a corpus. Corpus is generally a large collections of written texts here in our case farmer's queries that will be used for calculating the semantic similarity. This is always a onetime activity which will be saving time when user wants to get answers to multiple query.

The TF-IDF model is based on this idea which we have divided in below steps:

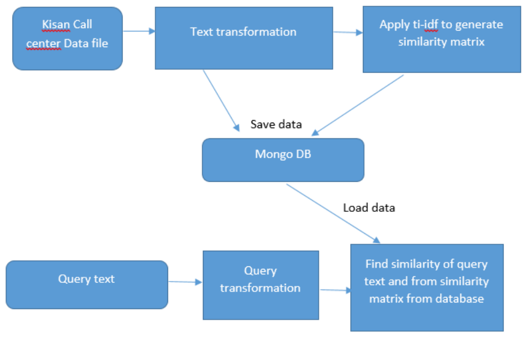

Kisan Call center Data file
Text transformation
Apply ti-idf to generate similarity matrix
Save data
Mongo DB
Load data
Query text
Query transformation
Find similarity of query text and from similarity matrix from database

Figure 2 – Base TI-IDF model.

Our model is based on this idea which we have divided in below steps:

- Corpora generation: The kisan call centre data file is tokenised to remove the stopwords and the words appearing once. After this we will be creating a genism dictionary and this data is saved using Mongo DB. To make this corpora more efficient we will be converting the dictionary text to bag-of words format and save this in mongo database in MM (market matrix) format. Now we have the corpora ready to be used for further analysis. This corpora generation is one time activity and for huge dataset this will be saving lot of time if user has to find answers to multiple queries.
- Finding Unit Vector Matrix: After creating the corpora, we will loading it and then will create the ti-idf model. With the ti-idfthe unit matrix will be calculated and again saved to the mongo DB. This will be used for generating the similarity matrix for LSI and LDA models separately, this similarity matrix once generated will be saved in the database, and for getting the answers for queries the entire dataset will not be queried but only these small saved matrix will be sufficient for getting query answers. This is also one time activity and it will also save time for subsequent query.

- Query Processing: In this step we will be loading the unit vector matrix from mongo db. Feed a query to the model. This query which will be fed to the model will also be converted to vector space using the dictionary stored in the mongo DB. After this the query vector, we need to change the query text and get the similar queries from the similarity matrix.

3.1 TI-IDF with LDA:

Once we have TF-IDF vectors next step is to use LDA in pipeline with this. LDA converts this Document-Term Matrix into two lower dimensional matrices, which helps in getting the query results faster since dimensionality is reduced. LDA is a probabilistic generative model which assumes that every document is distribution over topics and every topic is a distribution over words. Each word in a document is generated by first sampling a topic from the topic-distribution associated with the document and then sampling a word from the word distribution associated with the topic. Thus, given a corpus, LDA tries to find the right assignment of topic to every word such that the parameters of the generative model are maximized [7].

Topic Modelling is a widely used technique in information retrieval, data mining because of we can easily represent a large corpus by using small number of latent topics. In the real world data we may find large dataset which can be leading to some few topics, in these kind of scenarios these models are proved to be very effective. LDA algorithm which we are experimenting with our work is also based on topic modelling and the required results can be obtained in five different steps.

1. **Defining the number of topics to be grouped:** In this step we need to define number of topics which we need to the corpora to be divided. This can be done by using an informed estimate depending upon the previous analysis or by using trial and error method
2. **Assigning every word to a temporary topic**: This is work of the algorithm to assign every word of the corpora to a temporary topic. The topic assignment is according to Dirichlet distribution and it is done in a semi random manner. Since this step we are adding to a temporary topic this may happen that same word can be assigned to different topics. Also this step has already taken care of removing the stop words like “the”, “and”, “my” and these are not assigned to any topics.
3. **Updating the topic assignment:** The algorithm now will update the topics by looping through the document and checking for each word in the document to update the topics assigned to it. For each word the topics are updated based on two criteria.
    a. How frequent is the word across the document?
    b. How frequent is the topics spread across the document.
4. **Topic to Document mapping**: After the topic creation is done following steps we will follow to process a new query
    a. Find the threshold, let's set the threshold to be 1 clusters, to prove that the threshold is sane, we average the sum of all probabilities.
    b. Now we need to find out the Lda Corpus and document corpus and see if the threshold exceeds to the threshold we calculate, or we need to calculate the max probability out of all topic probability and group accordingly
5. **Answer retrieval**: Convert new query to vector, pass the vector to LDA model which in turn will return the probability for the document to be present in a particular topic. We will return the topic which has highest probability. After getting the topic we will display the documents belonging to that topic.

3.2 TI-IDF with LSI:

Latent Semantic Indexing (LSI) is a method for discovering hidden concepts in document data. Each document and term (word) is then expressed as a vector with elements corresponding to these concepts. Each element in a vector gives the degree of participation of the document or term in the corresponding concept. The goal is not to describe the concepts verbally, but to be able to represent the documents and terms in a unified way for exposing document-document,

document-term, and term-term similarities or semantic relationship which are otherwise hidden[7].

After we have decided to go for LSI, we will be following the below given steps for preparing the model:

1. **Number of features to be kept after decomposition**:The number of features can be decided based on the previous analysis or we can go for trial and error method. We can pick the best results by trying different estimates and hence lock on the number of features to be used for the algorithm.
2. **Building LSI model:**Prepared TF-IDF vectors will be fed in pipeline to the LSI, then this LSI will decompose the bigger TF-IDF vector to smaller vectors of size N*K, where N= number of documents and K= number of components.
3. **Build the Unit Matrix:** After generating the LSI model we will store this as unit matrix. The same will be applicable for all the documents in the corpora.
4. **Query Processing:** This step will be used to get the query from the user. This query will be converted to query vector, this will be based on the same model which we have generated. On basis of this query vectors we will be getting the query answers.

IV. EXPERIMENT AND ANALYSIS

We have generated first the corpora for the TI-IDF, LSI and LDA models. This is one time activity hence if user wants to fire multiple queries, he will save tremendous amount of time by using corpora instead of running the query on the entire dataset.

For our dataset which contains more than 1 lakh entries, for building entire corpora the time taken was 1:37:23.083620. This is one time activity, henceforth this time will not be taken when we fire any query to get the results.

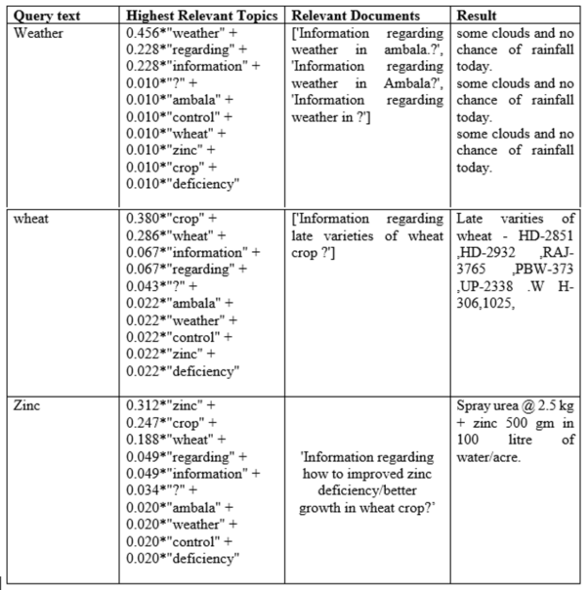
4.1 Results with TF-IDF/LDA model:

For our analysis we have used one year data for all district of Haryana, India. That data is fed to the model and corpora is generated. The model gives back 2 sets of information. One is topic to document mapping and topic to tokens mapping. We feed the new query to the model. Say a person wants to know the answers for his particular query, he has to give the query to the model, and this model will find the most relevant topic for the query given. From the relevant topic we get the associated query documents. As we have answers for those documents those corresponding answer can be displayed as per the relevance. We have fired different queries to the corpora and its results we are giving as below.

This works well with all size of corpora, but if we have really big size corpora this model will help in getting the query faster, since this model will break the corpora in different topics hence whenever a query is fired we will not be traversing through whole corpora but with the highest relevant topics and its documents. This way it's saving lot of time and also giving better result.

Like every model this model has one disadvantage that it takes huge time for building the topic-document-token matrix. With 1 lakh 18 thousand dataset size, this model takes around 1 hour and 40 minutes for building the topic-document-token matrix. But since its one time activity, once done we will not be creating this again and again this time will be saved.

This model query processing is very fast for getting a query answers for a particular query for this big corpora it just takes 2 seconds to find the relevant documents and the answers.

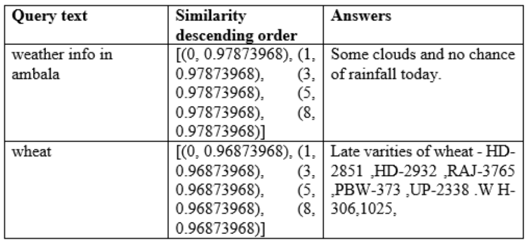


Query text	Highest Relevant Topics	Relevant Documents	Result
Weather	0.456*"weather" + 0.228*"regarding" + 0.228*"information" + 0.010*"?" + 0.010*"ambala" + 0.010*"control" + 0.010*"wheat" + 0.010*"zinc" + 0.010*"crop" + 0.010*"deficiency"	['Information regarding weather in ambala.?', 'Information regarding weather in Ambala?', 'Information regarding weather in ?']	some clouds and no chance of rainfall today. some clouds and no chance of rainfall today. some clouds and no chance of rainfall today.
wheat	0.380*"crop" + 0.286*"wheat" + 0.067*"information" + 0.067*"regarding" + 0.043*"?" + 0.022*"ambala" + 0.022*"weather" + 0.022*"control" + 0.022*"zinc" + 0.022*"deficiency"	['Information regarding late varieties of wheat crop ?']	Late varities of wheat - HD-2851 ,HD-2932 ,RAJ-3765 ,PBW-373 ,UP-2338 .W H-306,1025,
Zinc	0.312*"zinc" + 0.247*"crop" + 0.188*"wheat" + 0.049*"regarding" + 0.049*"information" + 0.034*"?" + 0.020*"ambala" + 0.020*"weather" + 0.020*"control" + 0.020*"deficiency"	'Information regarding how to improved zinc deficiency/better growth in wheat crop?'	Spray urea @ 2.5 kg + zinc 500 gm in 100 litre of water/acre.

Figure 3 – Result TI-IDF/LDA model

4.2 Results with TF-IDF/LSI model:

For our results we are using the same dataset as we have used for LDA model. This model will generate a unit matrix, which is getting calculated from unit vectors. This is one time activity, but it's not time consuming it takes 30 seconds to generate the corpora for the above said dataset. If a user wants to get answers of a query he has to give the query text, this text will be converted to the query vector, this vector will be compared against the saved unit vector for getting a similarity index. On basis of this similarity index the query answers will be picked from unit vector and displayed to the user. This model takes around 2.5 to 2.6 seconds for finding a query answer.



Query text	Similarity descending order	Answers
weather info in ambala	[(0, 0.97873968), (1, 0.97873968), (3, 0.97873968), (5, 0.97873968), (8, 0.97873968)]	Some clouds and no chance of rainfall today.
wheat	[(0, 0.96873968), (1, 0.96873968), (3, 0.96873968), (5, 0.96873968), (8, 0.96873968)]	Late varities of wheat - HD-2851 ,HD-2932 ,RAJ-3765 ,PBW-373 ,UP-2338 .W H-306,1025,

Figure 4 – Result TI-IDF/LSI model

V. FUTURE WORK

The model has been tested for one district dataset, which we will be testing for whole State and then whole country dataset. This will help us to tune the model for bigger dataset. We will also fine tune the model using the time of query asked, which will help the agricultural sector to be prepared for the seasonal/monthly problems faced by the farmers of the country.

To make the model interactive this query result can be converted in to audio and played back to the user over the phone call for faster answer of the query using text to speech or vice versa. This will reduce the call time and also farmers may get their answers quickly.

We are planning to apply neural net based models (word embedding) for more contextual search as well. This will help if a user is not giving a full query to the system the system captures a misspelled query. This will make the model more efficient for getting better results from any form of query.

VI. CONSLUSION

India major population is earning their livelihood from agricultural sector. If they get help timely and quickly they can improve their productivity. Productivity increase in agriculture field will not only help the farmers and their family but also the people of country. This paper has presented one model for Kisan call center dataset. From the above experiment out of the two proposed models The TF-IDF/LDA seems to give better result if a user is trying to get multiple answers at a shot, this is helpful since we will not be generate matrix again and again, and the query processing will be based on the saved topics document mapping, which is a faster activity due to the fact that not the whole corpora is scanned again and again, but only a small matrix is scanned to get the answers. On the other hand the TF-IDF/LSI model scans the unit matrix which is for the all documents hence the traversal time for this is more, for getting multiple query answers it is not an efficient activity.

VII. REFERENCES:

[1] Kisan call center dataset, (data downloaded) https://data.gov.in/catalog/district-wise-and-month-wise-queries-farmers-kisan-call-centre-kcc-during-2017

[2] R. B. Yates, B. R. Neto, "Modern Information Retrieval" in , New York:ADDISON-WESLEY, 1999.

[3] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean.Efficient estimation of word representationsin vector space.ICLR Workshop, 2013

[4] Yuhua Li, David McLean, Zuhair A. Bandar, James D. OShea, and Keeley Crockett, Sentence Similarity Based on Semantic Nets and Corpus Statistics, Ieee transactions on knowledge and data engineering, Vol8, No. 8, August 2006

[5] genism : Corpora API documentation (online),https://radimrehurek.com/gensim/apiref.html

[6] MichelineKamber, Jian Pei, Jiawei Han, "Data Mining: Concepts and Techniques", 3rd Edition

[7] C. D. Manning, P. Raghavan and H. Schutze. Introduction to Information Retrieval, Cambridge University Press.