Using TF-IDF on Kisan Call Centre Dataset for Obtaining Query Answers

Sandeep Kumar Mohapatra,
Aricent Technologies Private Limited,
Bangalore,
India-560103
Email: sandeep01it22@gmail.com

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

Abstract - Getting semantic similarity in short texts plays an important role for many tasks in the field of information retrieval. This helps in getting search results, fetching answers to queries, building summary of documents etc. We present an approach for manually and automatically getting answers to the different problems and queries of the farmers for their day to day agricultural work. Using this approach, we can provide a query to the model, to find relevant questions asked for that query and their possible answers. We have first preprocessed the data and converted to a similarity matrix which we save in a database using mongoDb. By taking the saved data from database we trained the model to get the information of the query based on similarity between the sentences of the queries, and then the application will find the best possible answer according to the similarity. We will be using Term-frequency-inverse document frequency (TF-IDF) to find the similar queries. With TF-IDF, every word is given weight, the TF-IDF is measured by frequency the relevance is not taken into consideration for this model. This information can be used in trainings to boost call agent effectiveness and improve the customer experience. As an added bonus, more effective communication reduces handle time and operating costs.

Keywords: ti-idf, semantic, corpus, mongoDb, information retrieval.

I. INTRODUCTION:

Farmers of India lack in technical expertise hence they requires guidance of experts more than anything else for smooth working and higher productivity. Hence for providing technical updates relating to the agricultural works government of India has introduced Kisan call center (farmers help line) India on January 21 2004 to help the farmers of the country to get answers to their queries about the difficulties they face during their day to day agricultural works.

The Kisan Call Centre is a fusion of two separate and mutually exclusive technologies. One being the experts in agricultural field other being experts in information and communication Technology. Both have their own specialized persons of their respective domains. The kisan call centre software captures caller's details and the query and the answers to the queries provided by the kisan call centre team. These details are captured and given open access for the public by the government. The data includes district wise - month Wise details of queries asked by farmers and answers given by given by the kisan call centre executives.

In this paper we are trying to use the information retrieval techniques [2] to process past data for getting most common problems area-wise, crop-wise and problem wise analyses within the time-space framework and provides preventive and advance action solutions from both qualitative and quantitative aspects. Depending upon the questions asked by the farmers of a particular area we can get information of most common problems which are faced by the farmers of that area for particular crops. This information can be used to train the farmers to increase their yield and hence be more productive and also helps to identify pest attacks in any particular geographical area and the information collected is provided to the State Agriculture Department for taking suitable timely action through broadcasting on Kisan call centre itself.

The query text is in the form of short text segments and with different contexts. The dataset has queries related to farm inputs, pesticides, herbicides, high yielding varieties of seeds, livestock, policy issues, government interventions 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. Since we have number of contexts or we levels It is important to point specific textual similarity so that different levels and contexts can be segregated. This is required since different persons can ask same questions

978-1-5386-2459-3/18/$31.00 ©2018 IEEE

on different context and using different language, so our main focus is to identify all such text segment which mean almost the same, this will increase efficiency and accuracy of the model. This will help in getting informations which will be helpful to the farmers in multiple ways. Not only farmers but also the kisan call centre executives can also use this framework to give better solution for the queries of the farmers.

II. DATA COLLECTION AND PREPROCESSING:

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. This is huge dataset to work on.

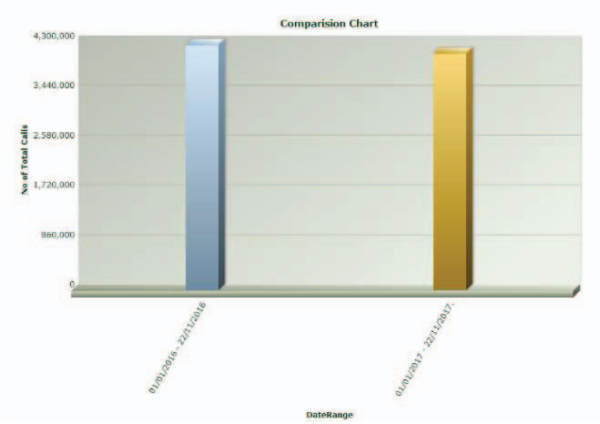

Figure 1 – total number of calls in India for a year

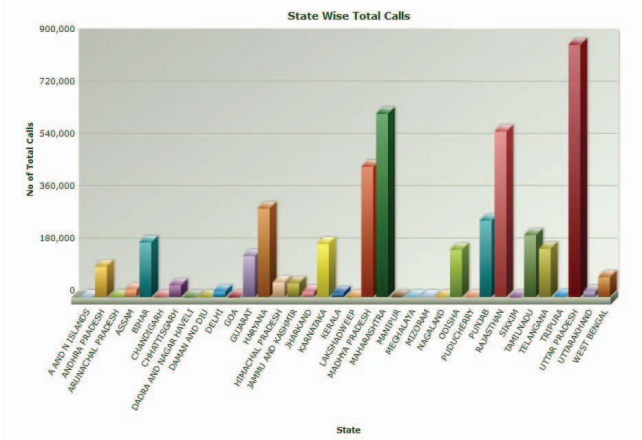

Figure 2– State wise number of calls

We have collected the data initially for districts of Haryana India, for whole year. For testing our model we have used the data for district Ambala, Haryana, India. [1]. The downloaded dataset was for different months in form of csv files. These files we have merged for a particular district to get data file for whole year. 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. 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 model for text transformation and ti-idf modelling.

III. SYSTEM MODEL:

Sentence similarity plays an important role in text mining. We are building a model on query texts represented 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.

It is very obvious that sentences with similar words will give more similarity score irrespective of considering the sentence meaning. For our model we will be getting the hierarchy by using ti-idf model and then we will get the word similarity by using the common syntax [3].

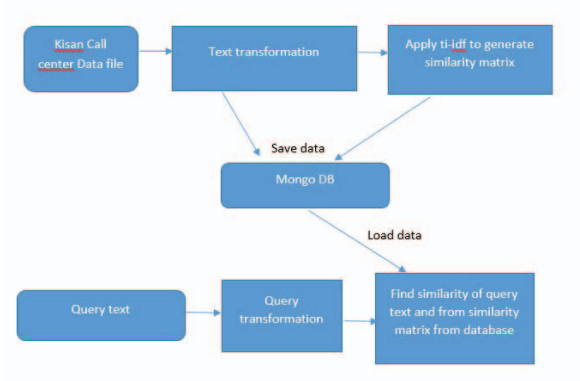

Figure 3 – Information retrieval model for KCC Data.

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].

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

1. 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.

2. Finding Similarity Matrix: Now the saved corpora model is loaded and ti-idf model is created. With this model we will be calculating the matrix similarity matrix and the calculated similarity matrix is again saved to mongo DB.

3. Query generation: In this step we will be loading the similarity matrix from mongo db. Feed a query to the model. This query which we will be feeding to the model will also be converted to vector space using the dictionary stored in the mongo DB.

The model is ready, we need to change the query text and get the similar queries from the similarity matrix. This idea is proved to be effective and helpful, we can easily cluster and find similar words in a huge corpus.

IV. EXPERIMENT AND ANALYSIS

For our analysis we have used one year data for Ambala district of Haryana, India. That data is fed to the model and corpora is generated. The similarity matrix is generated and then we feed the query to the model. Say for example person wants to know diseases affecting paddy crop. The user has to give the query as "paddy disease" this query text will find the similar queries relating to "paddy", "disease" and "paddy disease". Here we are using cosine similarity for evaluating the similarity measure for the query vectors with similarity matrix vectors. Cosine Similarity measure has been proved better for texts or sentences [6].

As per the above query below is the measures which we have obtained from the model. These are single query which we have captured for each query text. Depending upon the similarity index we can get the top queries asked by the farmers of that area, for a time frame or crop wise etc. This will be helpful to fight any common disease or problems in coming years for that particular crop.

Table 1 – Results table for query "paddy disease"

Query text	**Query from data file**	**Cosine Similarity index**
Paddy disease	how to control white /yellow leaf of **paddy** ?	0.260181
Paddy disease	HOW TO CONTROL FUNJAL **DISEASE** IN **PADDY** ?	0.643561
Paddy disease	how to control fungal **disease** in bottle gourd crop ?	0.215478
Paddy disease	information regarding how to control grasshoper in sugarcane ?	0

Table 2 – Results table for query "weather information ambala"

Query text	**Query from data file**	**Cosine Similarity index**
Weather information ambala	**Information** regarding **weather** in ?	0.497711
Weather information ambala	**Information** regarding **weather** in **ambala** ?	0.859959

The two tables here show similarity index for two sample queries which has been tested. We have given two queries “paddy disease” and “weather information ambala”, we found that the more matching words in the query text and the original similarity matrix has given better similarity index.

As per the above data we found that we can tune the query text to find better similarity index for more sentence matching, and less value of similarity index for lesser number of matching query texts.

While iterating to the whole data file we can then find the index for the higher similarity index queries and get the corresponding answers from the original dataset.

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

In 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. With this model an approach was presented to calculate sentence similarity score, the cosine similarity index. The more the value of index more accurate we will be getting the query from the big corpora.

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 representations in 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] Subhasini R, V. Jawahar Senthil Kumar, “Evaluating the Performance of Similarity Measures Used in Document Clustering and Information Retrieval”, Integrated Intelligent Computing (ICIIC) conference, 2010, Bangalore India