

A Report On
QUESTION ANSWER SYSTEM

Mtech Data Science

BY

Basu Verma

Under the Guidance of
Mrinal Kanti Sir

IIT PALAKKAD,
KERALA

CONTENT

1.) Abstract

2.) DBMS

3.) Information Retrieval

I. Pre-Processing

II. Bag of Word

III. TF_IDF

IV. Similarity

V. Conclusion

4.) FrontEnd

5.) References

My Part:-

In this project, we are group of three people. Their names are BASU VERMA, KULDEEP KUMAR, and ABHINAV KUMAR.

My contribution in this project is that I made the Information Retrieval part, which consists of cleaning and pre-processing of data, then stemming and then lemmatizing. After that, I computed the TF_IDF value of the each word and converted the sentences into the vectors of TF_IDF value. After that, I computed the cosine similarity of the vector, finds the similarity between the questions, and retrieved the answer. After that I made the frontend part of the project which was the GUI made from the tkinter module of python.

Abstract:-

The enormous dataset creates challenges to respond to the information. Information Retrieval is the process of responding to information from the dataset in text form or can be many different forms like image search, audio search, etc. Information Retrieval is an important domain of natural language processing (NLP). In simple words, IR means to search, mostly in text form. To build Information Retrieval model different techniques were used. Data structures and algorithms for proper and efficient search. For effective information retrieval pre-processing, Text Mining, Text feature extraction plays an important role. A user interface application was created and support the data management and data presentation functions described in it. A database was created for retrieval and an expert table for the new queries to store. The effectiveness of text processing is determined by the complexity and dimensionality reduction of the feature vector. In this paper, a TF-IDF and Cosine similarity approach was used. It combines both feature extraction and feature selection techniques for data mapping and retrieval, involving standard features for effective text mapping. The model provides an answer with proper accuracy from the dataset.

DBMS:-

Database Management System is a collection of interrelated data and able to access those data. Database changes over time. The collection of information stored in a database at a particular moment is called an instance and the overall design is called schema. Database application: DDL used to define data and DML used to update or change the dataset. The relational model uses a collection of tables both data & representation among data. In the project, we create a database having two tables one for the question-answer dataset used for storing data, for the user query domain, and the other for the new query section asked by the user, which is not available in the dataset. A trigger is a special type of stored procedure that automatically runs when a specific event occurs in the database server created by the administrator. The trigger is used when an expert provides an answer to a new question that was not available, data is appended to the main dataset using a trigger. The domain of the dataset are Beyoncé, natural resource and python.

We have used Maria DB as the database in our project. In this, we have stored all the Questions and Answers in the form of table named “q_a” and the expert table for the expert section. The dataset that we have used consists of total 186 questions and Answers.

```
MariaDB [(none)]> use issue_tracker_system
Database changed
MariaDB [issue_tracker_system]> show tables;
+-----+
| Tables_in_issue_tracker_system |
+-----+
| expert                          |
| q_a                            |
+-----+
2 rows in set (0.001 sec)

MariaDB [issue_tracker_system]> _
```

INFORMATION RETRIVAL:-

1.) PREPROCESSING:-

After storing of data in the form of table of column Questions and Answers in the Maria DB, we extracted all the data, converted that into a Data frame and then cleaned all the questions from the Questions column of data.

In cleaning of data what we did was, first removed all the stop words using the Corpus package from NLTK Library module. After that we removed all the special characters or symbols or punctuations from each word of the questions and strips all the white spaces or empty strings and changed all the words to the same case i.e. lower case.

After cleaning, we stems each word to its root word in order to overcome the singular and plural noun. For this purpose we used two types of Stemmer, these are Porter Stemmer and Lancaster Stemmer that we have imported from the NLTK stem Library module. By using both the stemmer on each word of our dataset, we found that the Porter Stemmer is more effective than Lancaster Stemmer with our dataset, so we used Porter Stemmer on all the words.

After stemming, we lemmatize each word in order to overcome the tenses in the sentence. For this purpose, we used WordNetLemmatizer that we imported from the NLTK.stem module.

We applied all these three process to all the questions available in the Questions column of our dataset and then stored the cleaned dataset to the database, which In this case is the Maria DB.

2.) BAG OF WORD MODEL:-

In our case of Information retrieval, we used Bag of Word Model for converting all the questions into the vectors of the cleaned word.

3.) TF_IDF:-

To give the proper weightage to all the words available in our dataset, there are various techniques available and from that, we have used here Term Frequency- inverse document frequency (TF_IDF) method.

Here TF stands for Term Frequency and IDF stands for Inverse Document Frequency.

$$Tf(t,d) = \frac{\text{Number of times a word appear in a document}}{\text{Total number of word in a document}}$$

$$IDF(t,D) = 1 + \log \left(\frac{N}{\text{Number of documents containing term } t} \right)$$

Where N denotes Total number of documents.

To ease the computing of similarity between the user input question and database questions, we represented the questions in the form of the vector of TF_IDF values.

For this, first we computed the TF value of all the questions and of the input question and form the dictionary in which word is the key and its corresponding TF value is the key value.

After that, we computed the IDF value of each word available in our Questions Dataset and stored the word as the key and its IDF value as the key value of the word in the dictionary.

Next, we computed the TF_IDF vector of user input questions using the function created to find the TF and IDf value.

Next, we created the term-document matrix of the database questions for the word available input questions.

4.) Similarity:-

Next, we computed the similarity between the user input question and the entire question available in the database.

For this purpose, again, there is lots of method available but out of all these, we use the cosine similarity method.

Cosine similarity measures the distance between the vectors using the inner product of two vectors. It usually measures the cosine angle between the two vectors and determines that whether two vectors are in the same direction or not by giving the cosine of the angle of which the value is in the range of 0 to 1. It is often used to measure the document similarity in text analysis. If the value is 1, then the texts are most similar that is they are in the same direction and for the value 0, it means that the two text are at the 90 degree, that is they are most apart from each other and thus the value in between 0 and 1 gives the corresponding similarity between the texts.

$$\text{Cosine Similarity} = \frac{\langle \mathbf{X}, \mathbf{Y} \rangle}{\|\mathbf{X}\| * \|\mathbf{Y}\|}$$

i.e., it's the ratio of inner product of two vector to the norm of each vector.

This gives the similarity and for the maximum similarity, we are showing the retrieved answer that is the answer corresponding to the max similar question from the database.

FRONTEND PART:-

For the front-end part of our project, we used tkinter module, which is the inbuilt library of python.

Tkinter is the standard interface to the Tk GUI toolkit. It is used for developing of graphical user interface (GUI). It provides a powerful object-oriented interface to the Tk GUI toolkit. It provides various controls, such as buttons, labels, entries and text boxes used in the GUI.

We have created two GUI using tkinter:-

- 1.) For the User to enter the question in the space provided.

In this portal, the user will write the question under the domain, which is Beyoncé, Natural resource and Python.

The question and the answer corresponding to the most similar to the input question from the database will appear just below the user question.

If any question is not present in the database, then that question will be added to the list for the expert to answer and a message will be shown stating that the “Answer not found in the database and expert is notified.”

QUESTION ANSWER SYSTEM

ASK YOUR QUESTION

domain is 'Python', 'Natural Resources' and 'Beyonce'

Difference between list and tuples

SEARCH

Database Question

What is the difference between list and tuples in Python?

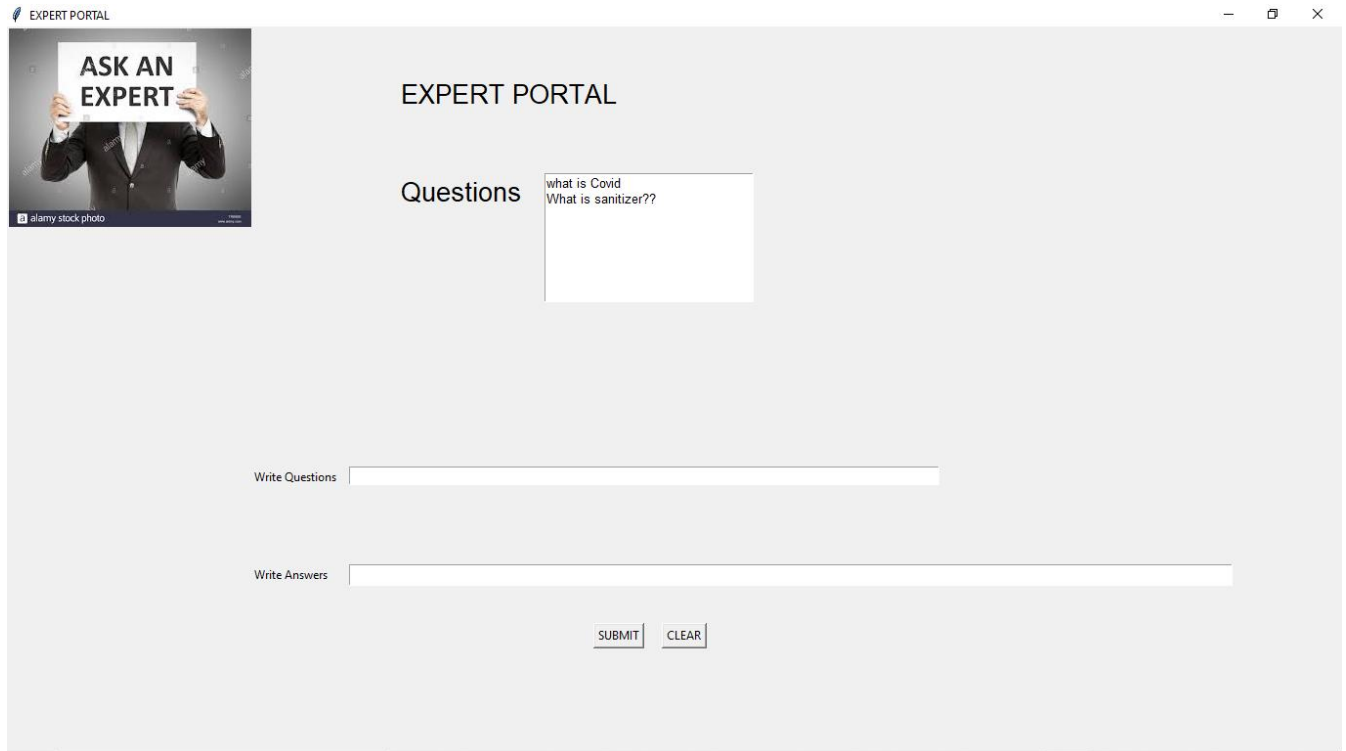
Retrieved Answer

Lists are mutable, but tuples are immutable.

2.) Expert Portal:-

This portal, we created for the expert. All the questions that is available for the expert to answer will be shown at the top. After that If, the expert will write the answer of a question then that question and answer will be saved in the database.

Next time when any user will write the same question, then the answer provided by the expert will be shown to the user.



The screenshot displays the 'EXPERT PORTAL' web application. On the left, there is a thumbnail image of a person holding a sign that says 'ASK AN EXPERT'. The main content area is titled 'EXPERT PORTAL' and features a 'Questions' section. Below this, there is a text input field containing the text 'what is Covid' and 'What is sanitizer??'. Further down, there are two more text input fields labeled 'Write Questions' and 'Write Answers'. At the bottom right, there are two buttons labeled 'SUBMIT' and 'CLEAR'.

CONCLUSION:-

In this paper, we take a dataset and use data retrieval for accessing information from the dataset, it is fast and accurate.

we use TF- IDF and the Cosine similarity technique in retrieving data. Structural information is represented by vector token form leading to textual information. Textual information is represented by the main root vocabulary to cause relationship similarity between words. The system is constituted by indexing.

REFERENCES:-

- 1.) <https://janav.wordpress.com/2013/10/27/tf-idf-and-cosine-similarity/>
- 2.) <https://dev.to/coderasha/compare-documents-similarity-using-python-nlp-4odp>
- 3.) <https://www.youtube.com/watch?v=ZxR38An5TQE&t=789s>