

ONLINE FORUM USING TEXT SIMILARITY

J COMPONENT PROJECT REPORT

Winter 2019-20

Submitted by

VANDITA PANDEY (17BCE0711)

ALISHA SHAH (17BCE0930)

NAVYA RG (17BCE2416)

in partial fulfillment for the award of the degree of

B. Tech

in

Computer Science and Engineering



Vellore-632014, Tamil Nadu, India

School of Computer Science and Engineering

May, 2020

I. ABSTRACT

In the modern day scenario, we have unlimited resources, but fail to find the most accurate answer to our queries. This is because there are multiple answers for similar questions with some variation, leading to a pretty different answer than our expectations. Thus, in order to find the most accurate answer, we need to ensure limitations of redundant questions, such that the best answers can be found with a simple click under that question. Question-answering or also called QA is an important area of research in natural language processing (NLP) with a long history of people trying to find the optimum way to work with it. A lot of people rely on data on online forums to seek solutions for their queries, instead of relying on search engines. This gives them a chance to understand the solution from the point of view of the people with the same question, thus resulting in it being more appealing to the user. Word2Vec strategy has been adopted in this where it takes input as a huge corpus of text and produces a vector space typically of several hundred dimensions. Thus, we present information retrieval forums using text similarity in natural language processing that makes use of forum data to analyse the question and provide the most optimal answer and desired search result to the question search. The main aim of our project is to have unique questions that can be answered by people who have knowledge about it.

II. INTRODUCTION

Our forum lets a user post a question and uses text similarity to identify if a similar question is already present or not. If it does, it forbids the user to post it. This is because a lot of duplicate questions may be asked in an online forum which will lead to multiple questions with the same meaning with many of them left unanswered. To avoid it, we use text similarity to identify questions that are being repeated and do not allow the user to post it and differentiate them from the unique questions. If a user posts a question that is similar, he will get suggestions of already existing closely-related questions and will help the user to find an answer easily. A Word2Vec model is used that is a shallow, neural network of 2 layers which is trained to rebuild linguistic contexts of words.

III. ARCHITECTURE DIAGRAM

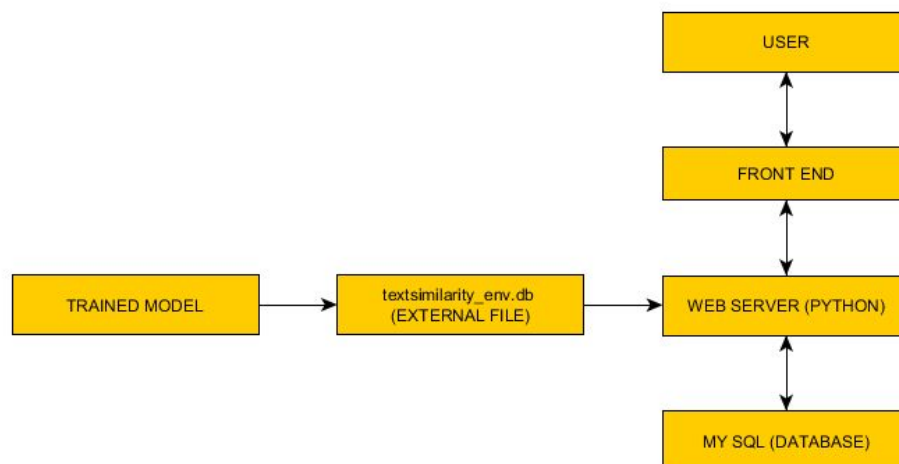


Fig 3.1: Architecture

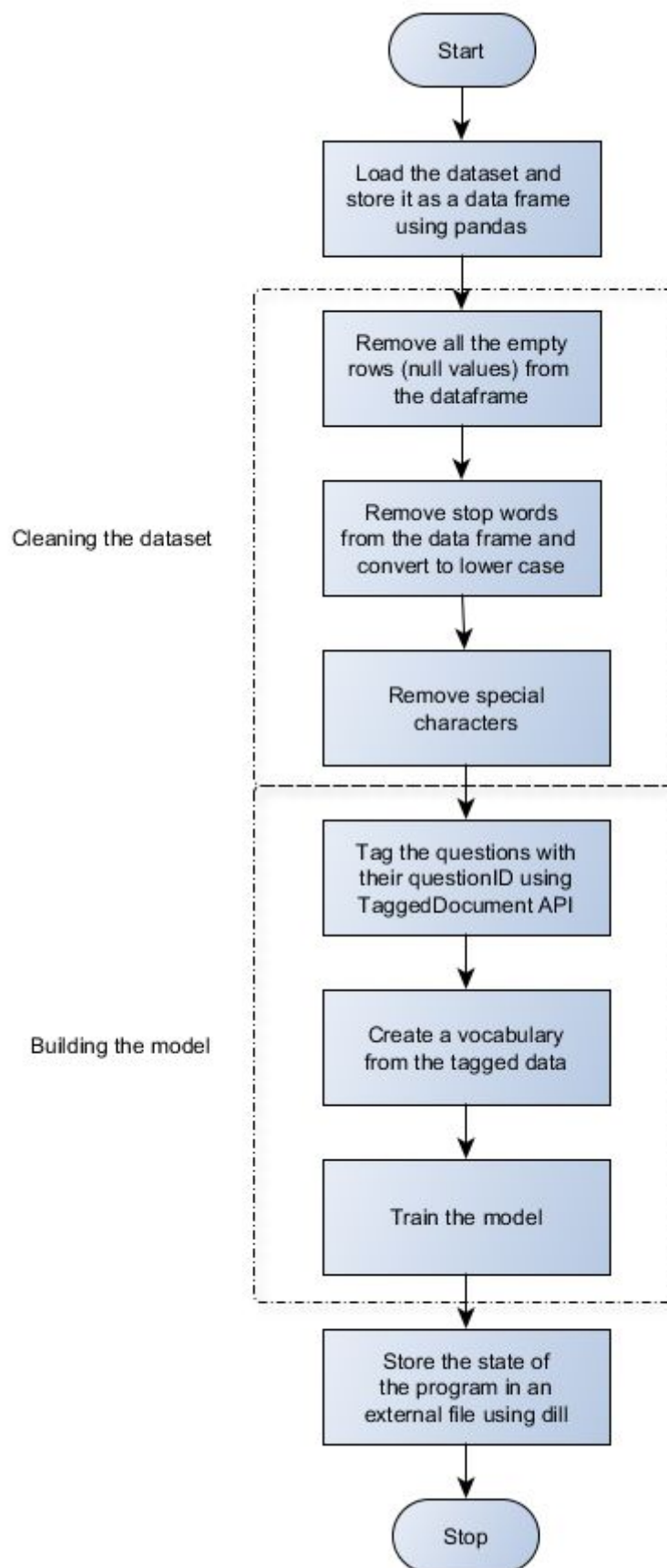


Fig 3.2: Flowchart for training the model

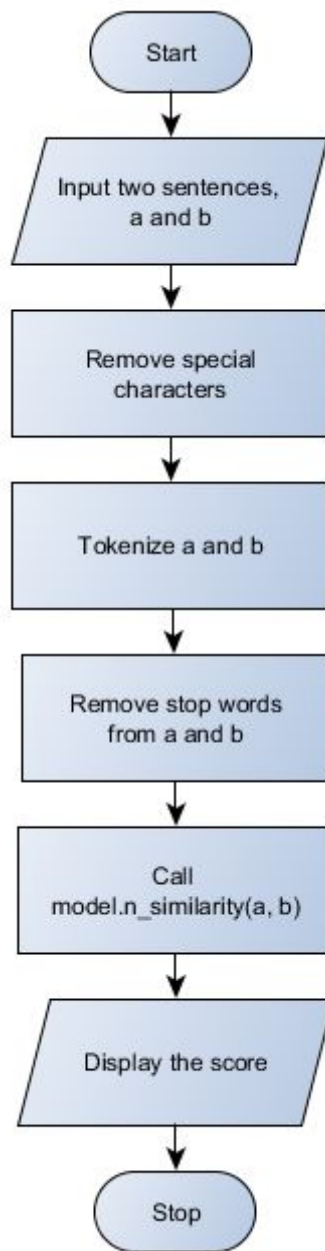


Fig 3.3: Flowchart for calculating the similarity between two sentences

IV. BACKGROUND STUDY

The accuracy of the similarity is largely dependent on an adequate amount of data and needs advanced analysis using natural language processing.

In the first paper, an approach towards improving document similarity calculation using cosine similarity was performed between candidates and was applied to movie reviews and sight-seeing locations. To improve the accuracy of the similarity assessment, sufficient data must be collected and advanced analysis is required through natural language processing. In

particular, the accuracy of the similarity is said to be improved only if labor is added to the preprocessing, such as selecting a stopword in natural language processing.

Using documents vectorized by TF-IDF or Doc2Vec, the cosine similarity of a document is obtained by the following equation:

$$\cos(v_u, v_i) = \frac{v_u \cdot v_i}{\sqrt{|v_u|^2 + |v_i|^2}}$$

Where v_u and v_i are vectors of the document. In the previous equation, the proximity of the angle formed by the vectors is expressed by taking a value from 0 to 1. A value closer to 1 means that the similarity between them is greater. [1]

The automatic answer to questions is a classic problem with natural language processing. The goal is to develop systems that can automatically answer a question like people. In a deep learning model for automatically answering questions, questions and answers are first integrated using probabilistic neural models. Next, a deep similarity neural network is created to find the similarity score for some answers and questions. So the best answer for each question is the one with the highest similarity index. Modern question-and-answer systems generally rely on a large amount of text in a knowledge source, which can be information on the World Wide Web or a structured knowledge base such as the free base. The main application of this model is the AT&T customer service chat system, where we use the proposed model to find the best answer in the database of old chat data and the confidence value of the correlation of the answer. selected for the determined question. If the confidence value exceeds a certain threshold, the client will restore it, otherwise the question will be sent to the human agent.[2]

Discussion forums on social media, and especially patient support forums, have repetitive questions and inquiries that reduce the likelihood that the community will respond. One possible reason is that patients / caregivers have certain situations to which they have asked questions, while there are similar cases that were answered in the same forum. Thus, to find the similarity of questions, cosine similarity is used.

Cosine similarity metric:

The angle between the query vector and each document's vector is calculated using: [3]

$$\text{similarity}(q_i, d_j) = \frac{V_{q_i} \times V_{d_j}}{|V_{q_i}| \times |V_{d_j}|}$$

Online discussion forums are websites where users can discuss and share their views on a variety of topics, from solving product problems to choosing resorts. This data can be used to improve access to the forum content, increase knowledge of the chatbot, integrate data from CQA (Community Question Answering) websites, etc.

All Part-Of-Speech tags were generated using the OpenNLP POS Tagger.

Similarity of Question: Use the candidate's cosine similarity to the answer and the respective question message after removing the stop words. [4]

With the volume of data available to the public, people use the Internet to search for answers to questions about specific information.

CQA is an online platform where users can ask or answer questions on various topics, depending on their interests. It uses a form of technical crowdsourcing in the community to determine the most appropriate answer to a question asked, based on a process that identifies questions or phrases on a question and answers websites with similar meanings. Advanced technologies such as artificial intelligence (AI) are now integrated into systems to improve system performance. It can be applied to question and answer systems to improve their effectiveness by providing adequate answers. In this way, CQA websites can become an essential knowledge base for the automatic provision of answers.

For clustering of similar questions, the distances between the sentences of the paragraphs are calculated based on the similarity of the cosine. [5]

One of the most studied areas is automatic coding based on the content of the articles. NLP methods in combination with machine learning have been used to support proper encoding. To get a quick overview of the scope of the problem caused by low-quality messages, we examined the cardinality of closed or deleted questions and changes to these values on the previous pages. Indicates the reasons why, for example, closure due to duplication or off-topic is not necessarily related to quality. In most cases, identifying these positions is relatively easy to automate.

There may be high-quality questions that are closed for another reason, since the question is not related to the topic. Therefore, recommendation systems should take a more nuanced and improved approach to deal with these exceptions. [6]

V. METHODOLOGY

We use the Word2Vec model to find the similarity between sentences.

Our approach first starts with importing data and cleaning it.

We download the csv file and drop any row with null questions by checking for null values.

Our next step involves removing any stop words.

A stop word is a commonly used word in a sentence, such as “a”, “an”, “the”, “in”, etc.

Next, we remove special characters (alpha-numeric or numeric characters that add extra noise to unstructured text).

Doc2Vec requires model training data in order to tag each and every question with a unique id, thus we tag the questions with their qid using TaggedDocument API, and before feeding the questions to the Model, we split the questions into different word and create a list of words for each and every one of them along with the tagging.

We then built the model and trained it.

We now iterate through each question pair and find the similarity between them using cosine similarity.

Cosine Similarity helps find the dot product between the vectors of two documents/sentences to find the angle and cosine of that angle to get the similarity.

With the help of cosine similarity, we find the similarity between question pairs and we set the threshold values:

0% \leq score $<$ 60% - No similarity

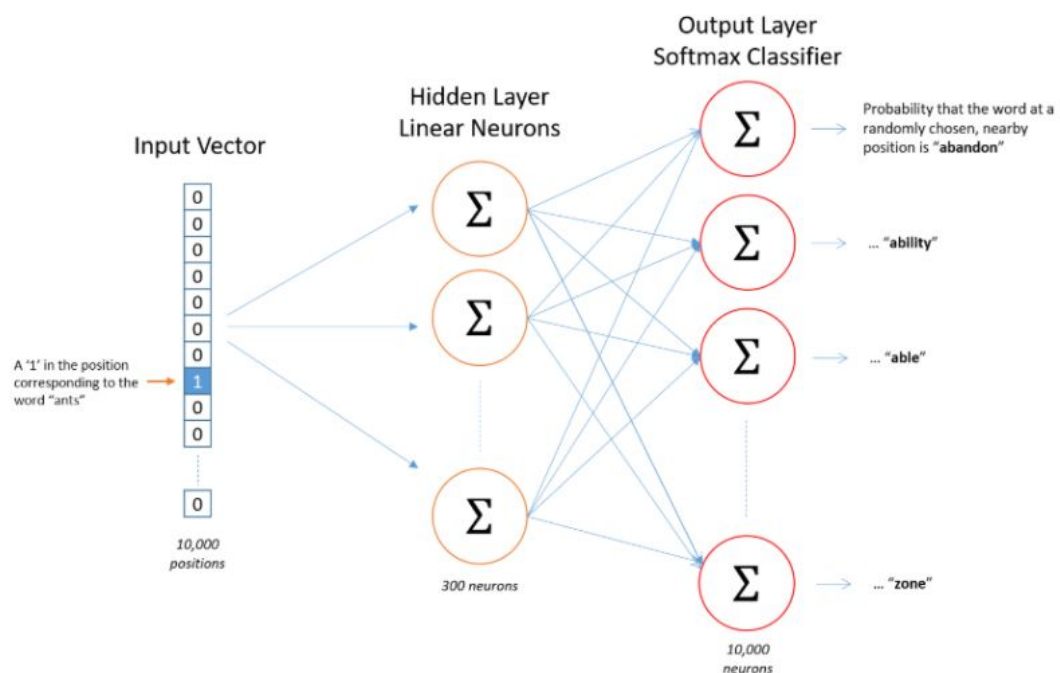
60% \leq score $<$ 90% - Slight similarity. Some level of relevancy is present.

90% \leq score \leq 100% - Very similar

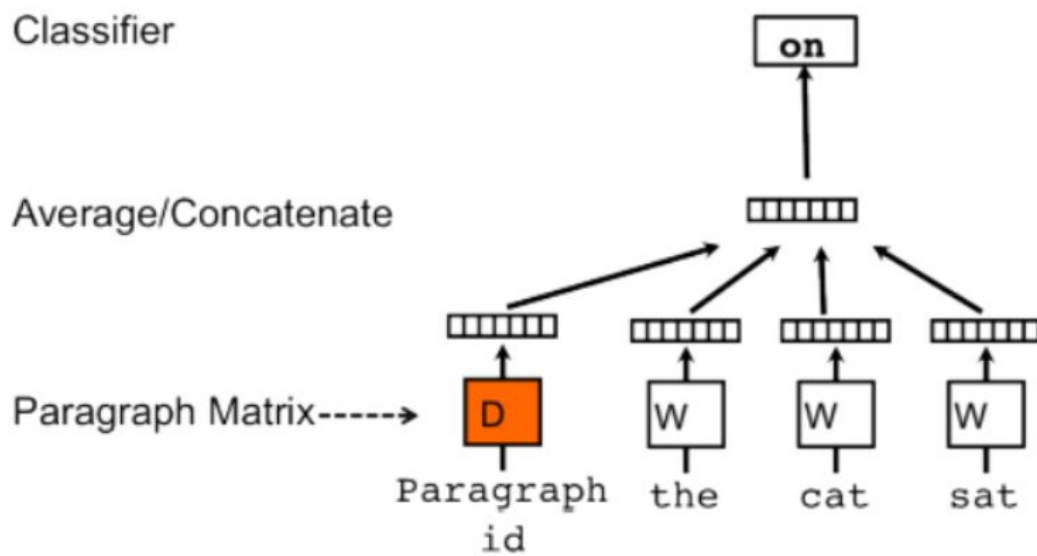
VI. PROPOSED MODEL

Word2Vec is a model that uses input as a large body of text and creates a vector space that generally has several hundred dimensions.

The basic assumption of Word2Vec is that two words that have similar contexts also have a similar meaning and therefore a similar vector representation of the model.



Doc2Vec is an extension of Word2Vec in which it makes numerical representations of documents or paragraphs instead of every word in the corpus. Thus, the vectors created by doc2vec are helpful for things like finding the similarity between paragraphs or documents.



VII. RESULTS AND DISCUSSION

The dataset we have used is the quora question pairs dataset. It contains 4 lakh pairs of similar questions. For example a pair of similar questions would be “My age is 20” and “I am 20 years old”.

id	qid1	qid2	question1	question2	is_duplicate
1	0	1	2 What is the step by step guide to invest in share market in india?	What is the step by step guide to invest in share market?	0
2	1	3	4 What is the story of Kohinoor (Koh-i-Noor) Diamond?	What would happen if the Indian government stole the Kohinoor	0
3	2	5	6 How can I increase the speed of my internet connection while usi	How can Internet speed be increased by hacking through DNS?	0
4	3	7	8 Why am I mentally very lonely? How can I solve it?	Find the remainder when $(23^{24})/24!$ is divided by 24,2	0
5	4	9	10 Which one dissolve in water quikly sugar, salt, methane and carbo	Which fish would survive in salt water?	0
6	5	11	12 Astrology: I am a Capricorn Sun Cap moon and cap rising...what d	I'm a triple Capricorn (Sun, Moon and ascendant in Capricorn) Wh	1
7	6	13	14 Should I buy tiago?	What keeps children active and far from phone and video games?	0
8	7	15	16 How can I be a good geologist?	What should I do to be a great geologist?	1
9	8	17	18 When do you use &, instead of &—?	When do you use "&" instead of "and"?	0
10	9	19	20 Motorola (company): Can I hack my Charter Motorola DCX3400?	How do I hack Motorola DCX3400 for free internet?	0
11	10	21	22 Method to find separation of slits using fresnel biprism?	What are some of the things technicians can tell about the durabi	0
12	11	23	24 How do I read and find my YouTube comments?	How can I see all my Youtube comments?	1
13	12	25	26 What can make Physics easy to learn?	How can you make physics easy to learn?	1
14	13	27	28 What was your first sexual experience like?	What was your first sexual experience?	1
15	14	29	30 What are the laws to change your status from a student visa to a	What are the laws to change your status from a student visa to a	0
16	15	31	32 What would a Trump presidency mean for current international n	How will a Trump presidency affect the students presently in US c	1
17	16	33	34 What does manipulation mean?	What does manipulation means?	1
18	17	35	36 Why do girls want to be friends with the guy they reject?	How do guys feel after rejecting a girl?	0
19	18	37	38 Why are so many Quora users posting questions that are readil	Why do people ask Quora questions which can be answered easil	1
20	19	39	40 Which is the best digital marketing institution in banglore?	Which is the best digital marketing institute in Pune?	0
21	20	41	42 Why are rockets look white?	Why are rockets and boosters painted white?	1
22	21	43	44 What's causing someone to be jealous?	What can I do to avoid being jealous of someone?	0
23	22	45	46 What are the questions should not ask on Quora?	Which question should I ask on Quora?	0
24	23	47	48 How much is 30 kV in HP?	Where can I find a conversion chart for CC to horsepower?	0
25	24	49	50 What does it mean that every time I look at the clock the number	How many times a day do a clock's hands overlap?	0
26	25	51	52 What are some tips on making it through the job interview proces	What are some tips on making it through the job interview proces	0
27	26	53	54 What is web application?	What is the web application framework?	0
28	27	55	56 Does society place too much importance on sports?	How do sports contribute to the society?	0

Fig 5.0: Dataset

The thresholds used in our project are given below:

Similarity score	Status
0% <= score < 60%	No similarity
60% <= score < 90%	Slight similarity. Some level of relevancy is present.
90% <= score <= 100%	Very similar

```
In [6]: from nltk.tokenize import word_tokenize

def similarity(a, b):
    a = a.lower()
    review_text = a
    review_text = re.sub(r'^A-Za-z0-9(),!.\?\'\"', " ", review_text)
    review_text = re.sub(r'\s', " ", review_text)
    review_text = re.sub(r'\ve', " ve ", review_text)
    review_text = re.sub(r'n\t', " t ", review_text)
    review_text = re.sub(r'n\re', " re ", review_text)
    review_text = re.sub(r'\d', " d ", review_text)
    review_text = re.sub(r'\ll', " ll ", review_text)
    review_text = re.sub(r",", " ", review_text)
    review_text = re.sub(r"\.", " ", review_text)
    review_text = re.sub(r"!", " ", review_text)
    review_text = re.sub(r"(", " ( ", review_text)
    review_text = re.sub(r")", " ) ", review_text)
    review_text = re.sub(r"?", " ", review_text)
    review_text = re.sub(r"\s{2,}", " ", review_text)
    a = review_text
    a_tokens = word_tokenize(a)
    a_filtered = [w for w in a_tokens if not w in stop]

    b = b.lower()
    review_text = b
    review_text = re.sub(r'^A-Za-z0-9(),!.\?\'\"', " ", review_text)
    review_text = re.sub(r'\s', " ", review_text)
    review_text = re.sub(r'\ve', " ve ", review_text)
    review_text = re.sub(r'n\t', " t ", review_text)
    review_text = re.sub(r'n\re', " re ", review_text)
    review_text = re.sub(r'\d', " d ", review_text)
    review_text = re.sub(r'\ll', " ll ", review_text)
    review_text = re.sub(r",", " ", review_text)
    review_text = re.sub(r"\.", " ", review_text)
    review_text = re.sub(r"!", " ", review_text)
    review_text = re.sub(r"(", " ( ", review_text)
    review_text = re.sub(r")", " ) ", review_text)
    review_text = re.sub(r"?", " ", review_text)
    review_text = re.sub(r"\s{2,}", " ", review_text)
    b = review_text
    b_tokens = word_tokenize(b)
    b_filtered = [w for w in b_tokens if not w in stop]
    score = model.wv.n_similarity(a_filtered, b_filtered)
    return score
```

Fig 5.1: Function for calculating similarity

The algorithm for calculating similarity given in Fig 5.1 is:

1. Convert both the sentences to lower case
2. Remove all the special characters from both the sentences
3. Tokenize both the sentences
4. Remove all the stop words from both the sentences
5. Calculate cosine similarity and return the score

```
In [13]: try:
        score = similarity("Which is the best mess in ladies hostel?", "In ladies hostel, which mess is good?")*100
        print(score)
    except:
        print("Error occurred")

99.62074160575867
```

Fig 5.2: Output produced by calculating the similarity

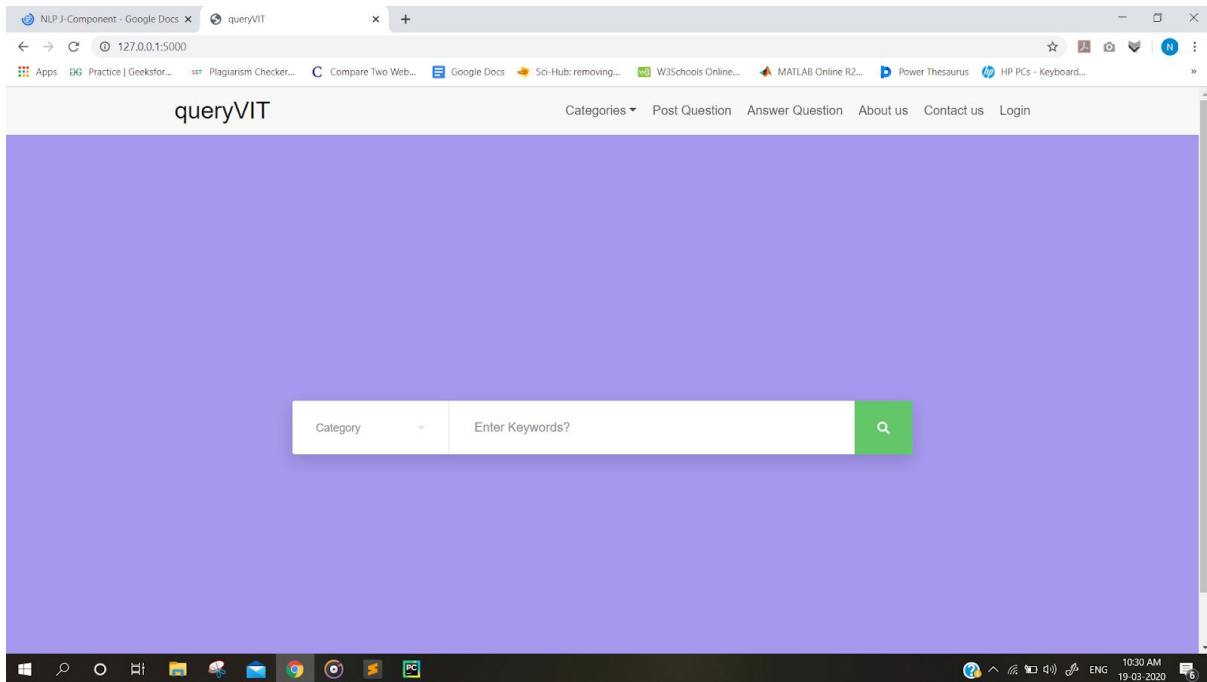


Fig 5.3: Homepage where questions can be searched for

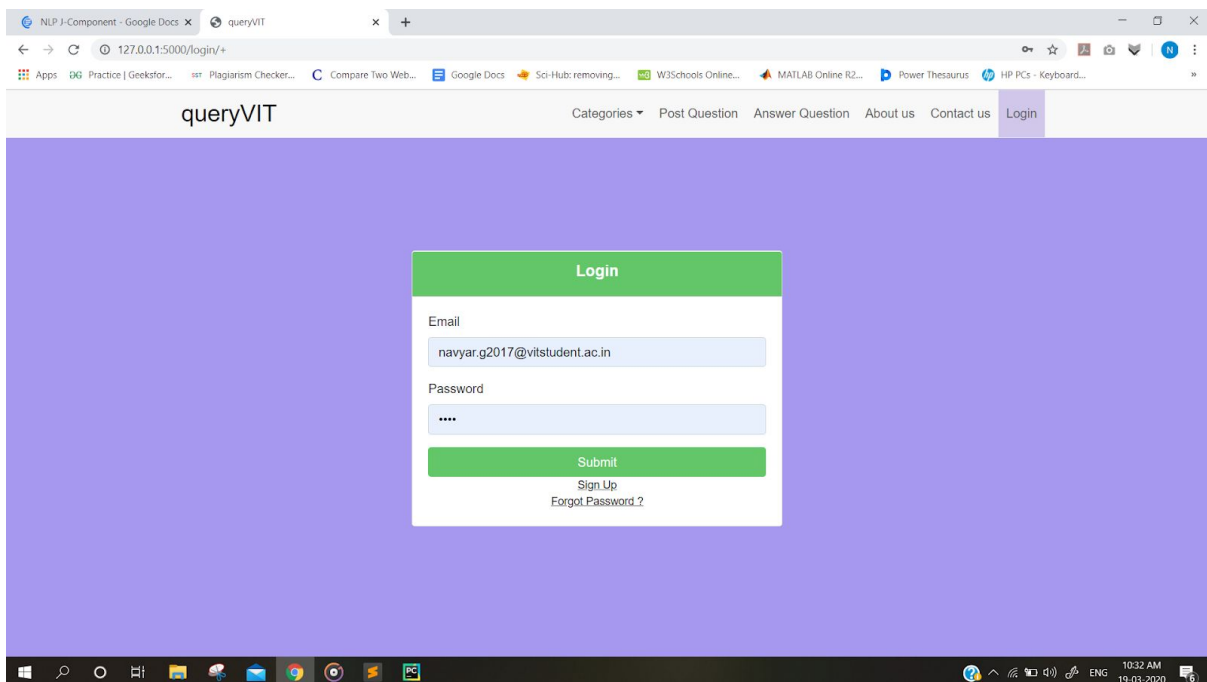


Fig 5.4: Login page

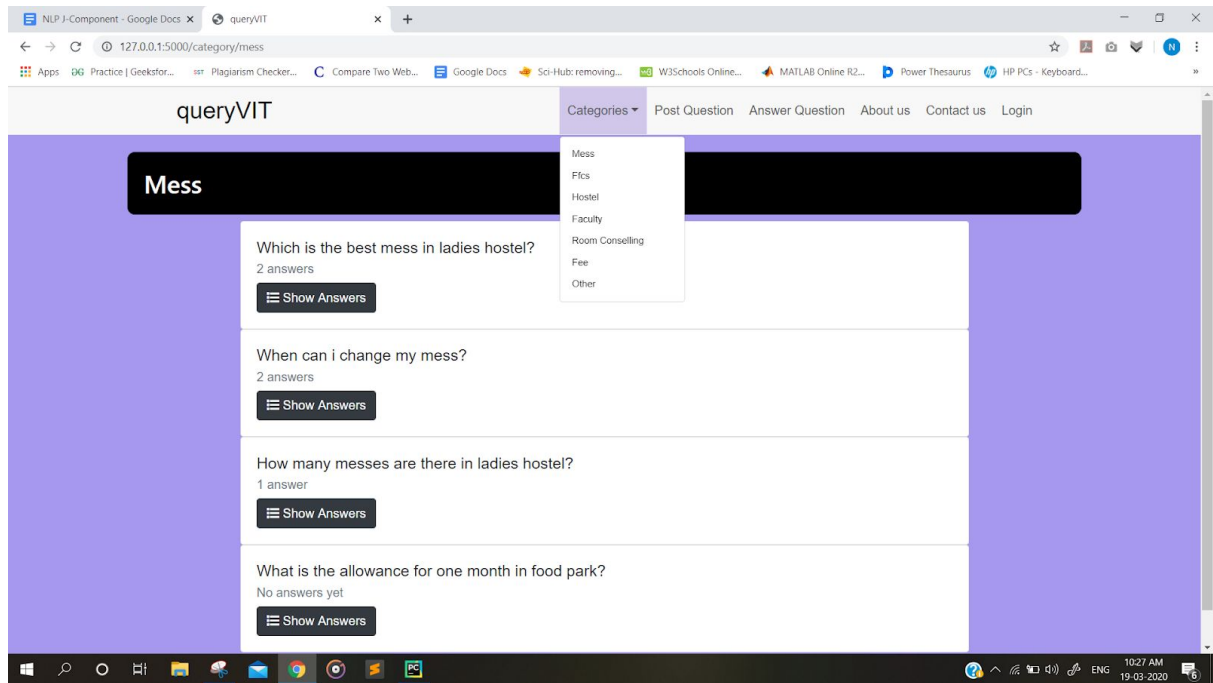


Fig 5.5: Category page where questions can be viewed according to the category

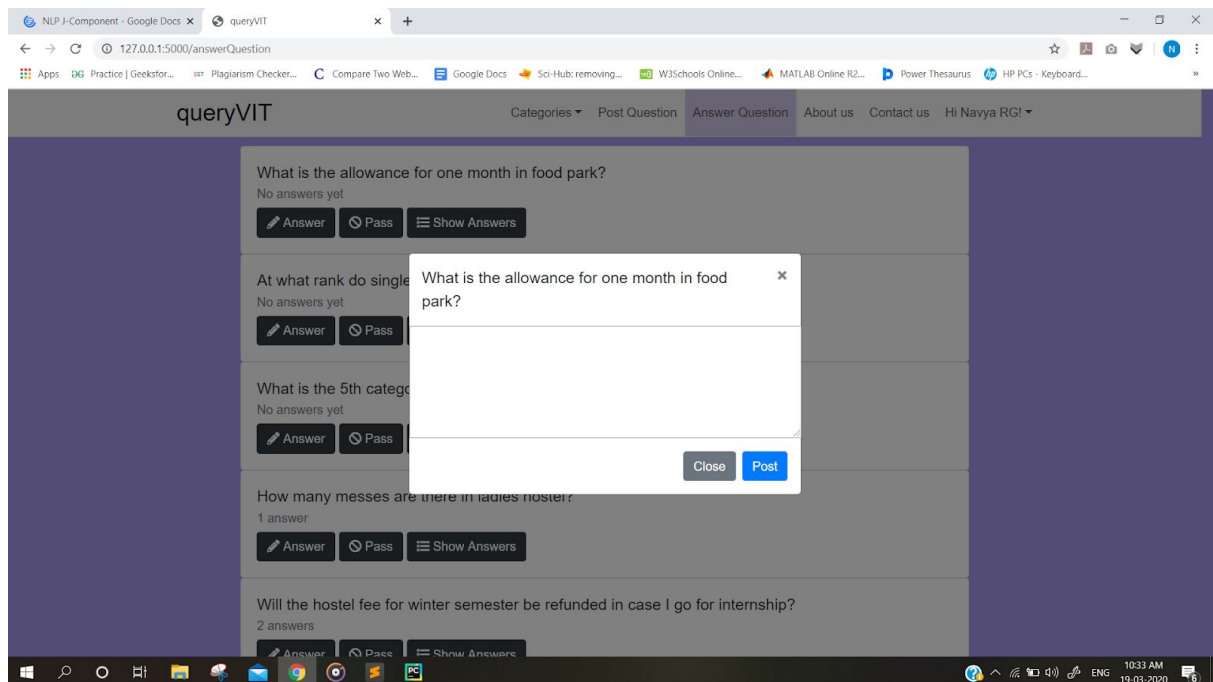


Fig 5.6: Answer questions page

The user is allowed to answer any question only if he/she is logged in.

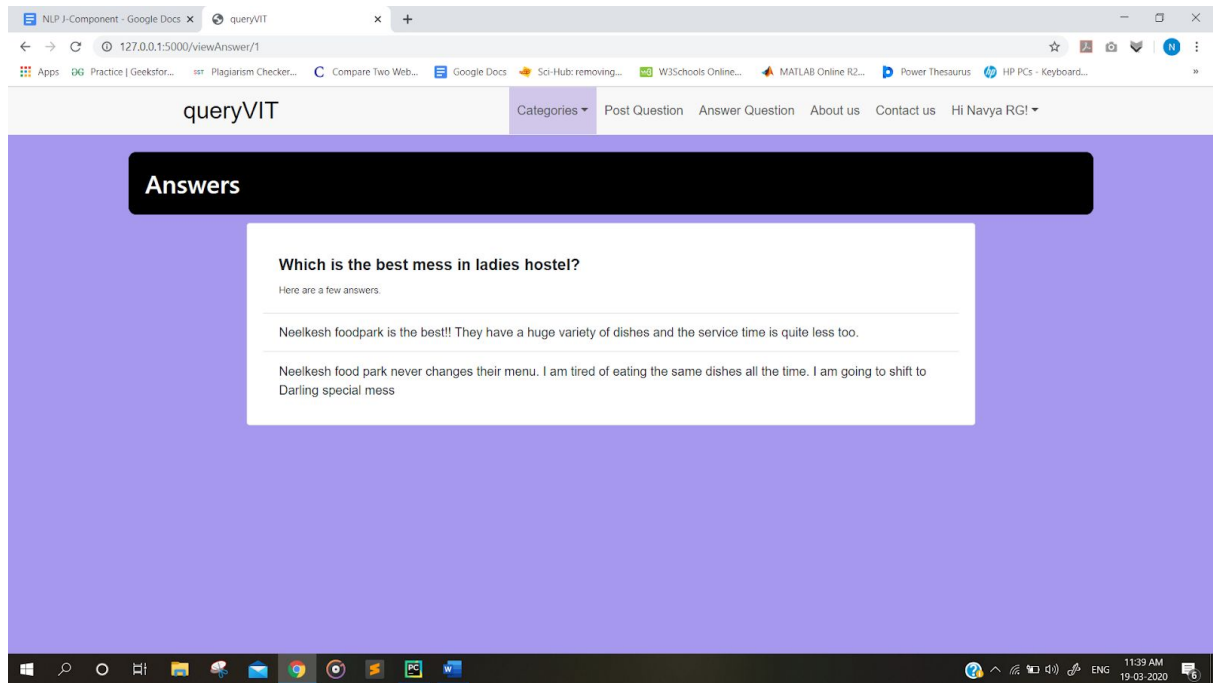


Fig 5.7: View answers page

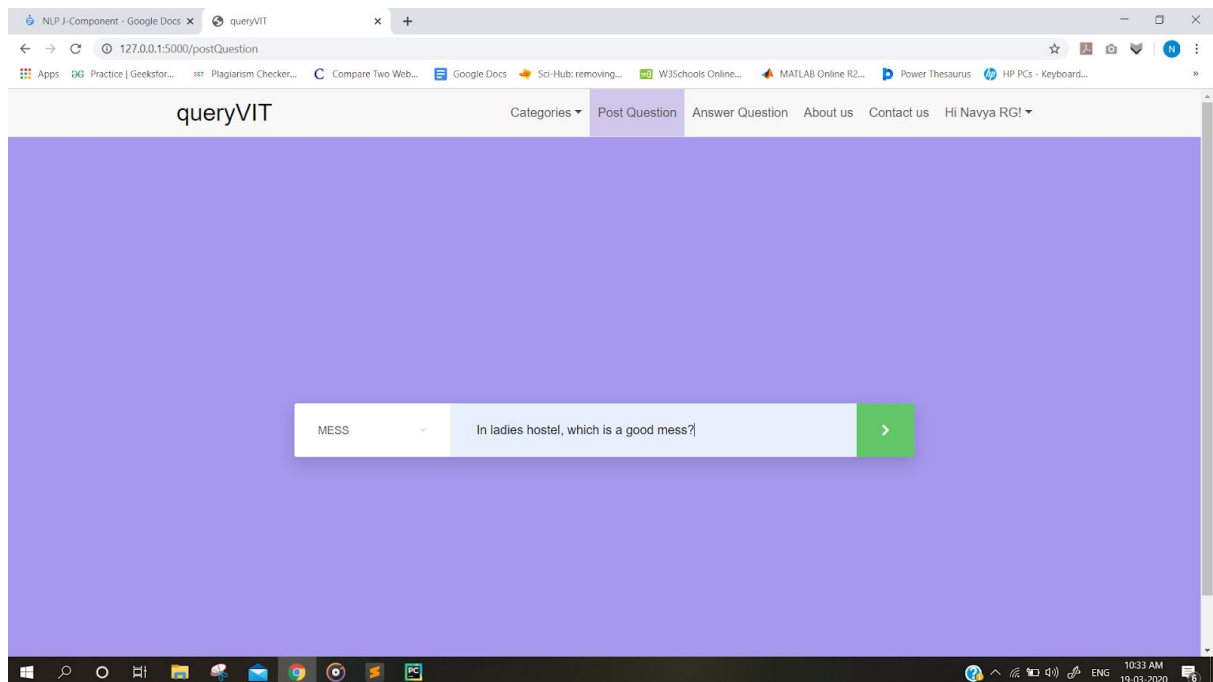


Fig 5.8: Post questions page

The user is allowed to post questions only if he/she is logged in.

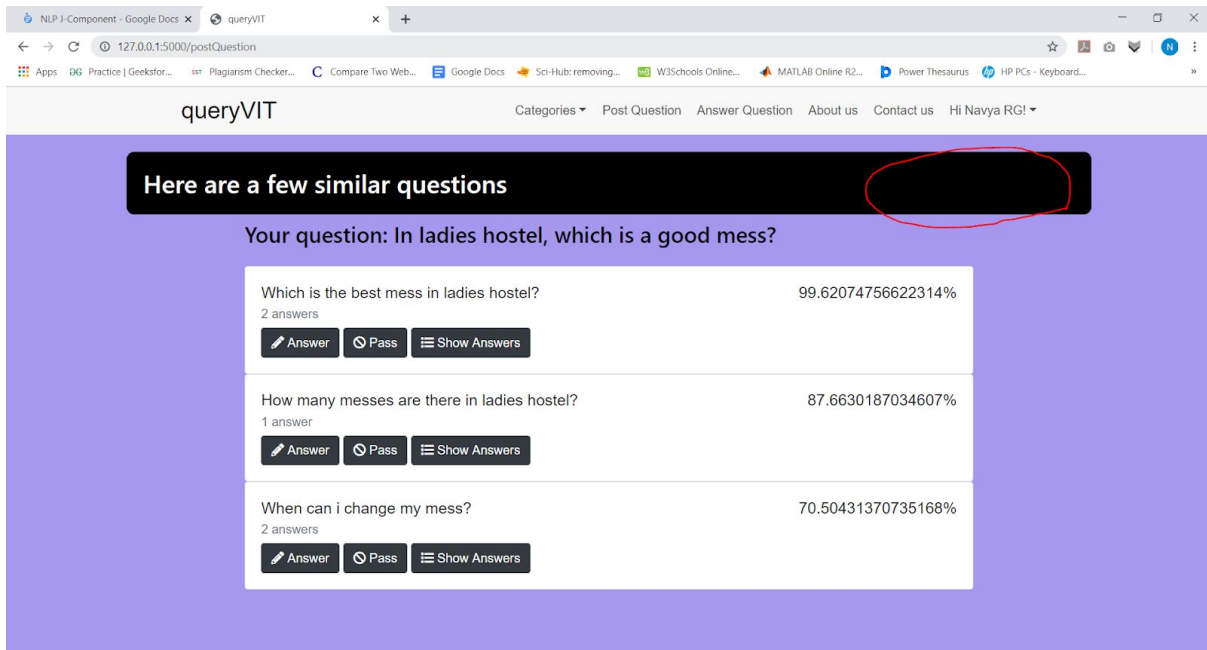


Fig 5.9: Similar questions page

In Fig 5.9 the question entered was “In ladies hostel, which is a good mess?” in the category “mess”. Since there are questions existing in the same category with similarity greater than 90%, the user is not allowed to proceed further (“Continue to post” button is not visible). At this point the only choice the user has is to view the answers of the most similar questions. In the similar questions page, the questions are sorted according to decreasing similarity. Apart from the questions which have a similarity greater than 90%, questions which also have a similarity of greater than 60% are displayed in case the user wishes to view them.

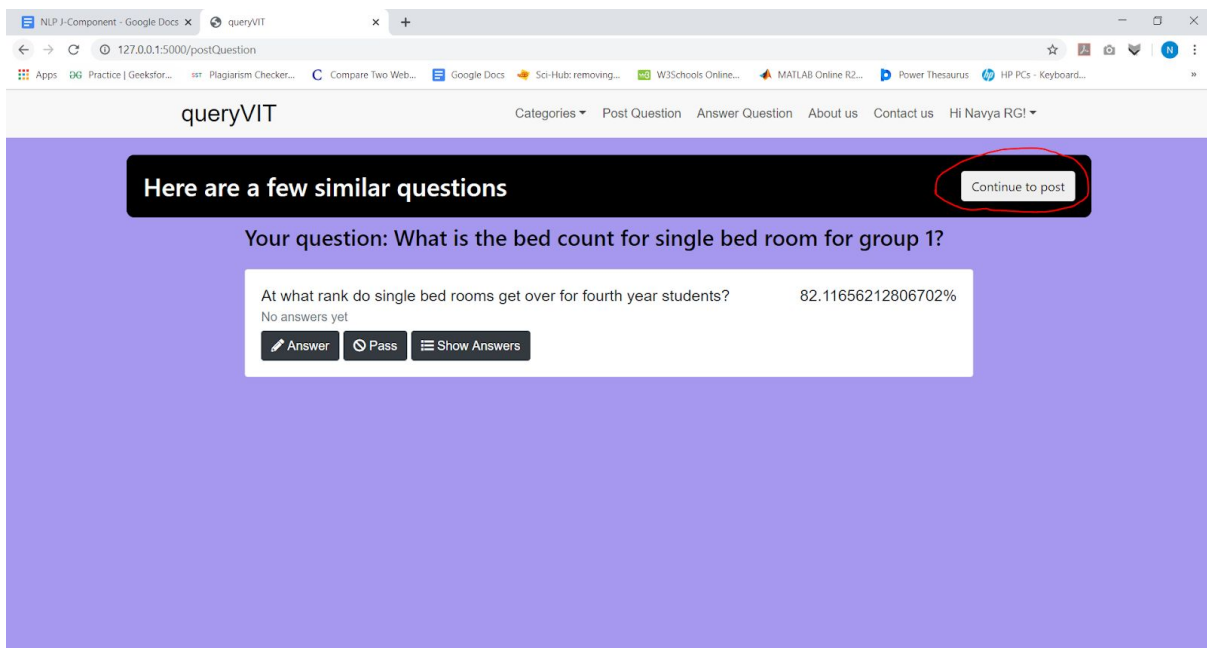


Fig 5.10: Similar questions page

In Fig 5.10, the question entered was “What is the bed count for single bed rooms for group 1?” in the category “room counselling”. Since there are existing questions with similarity greater than 60%, but lesser than 90%, the “Continue to post” button is visible. This is because the user’s question is not very similar to the existing questions and there is some difference.

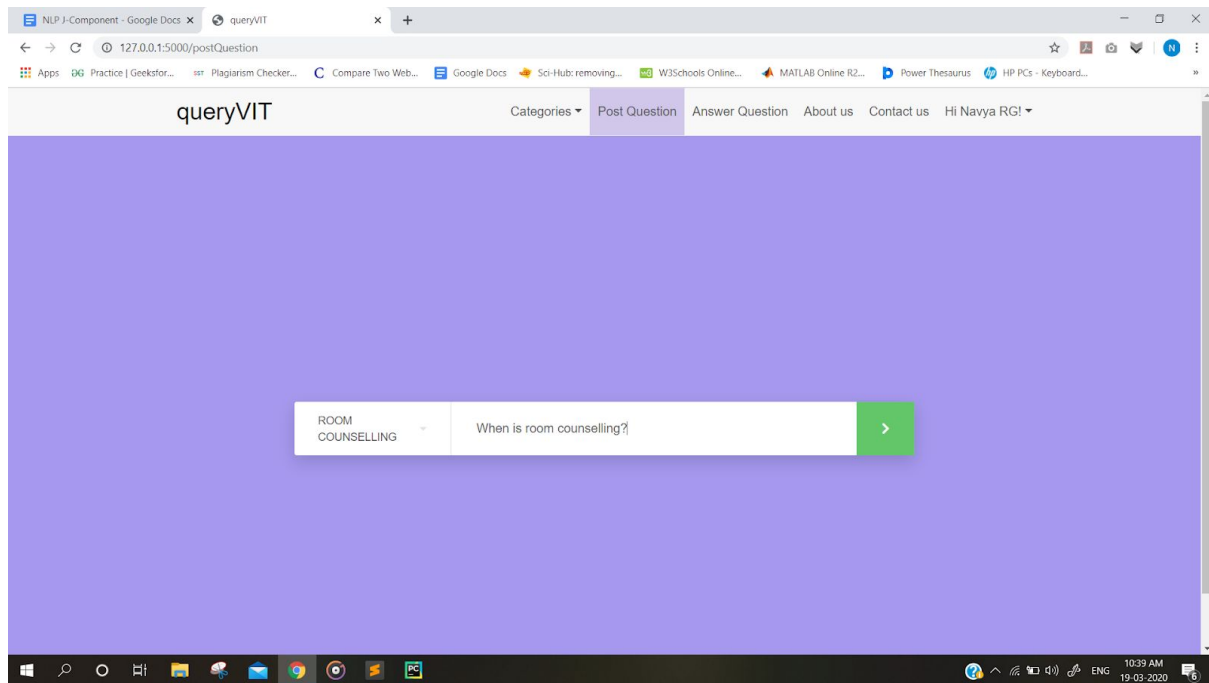


Fig 5.11: Post questions page

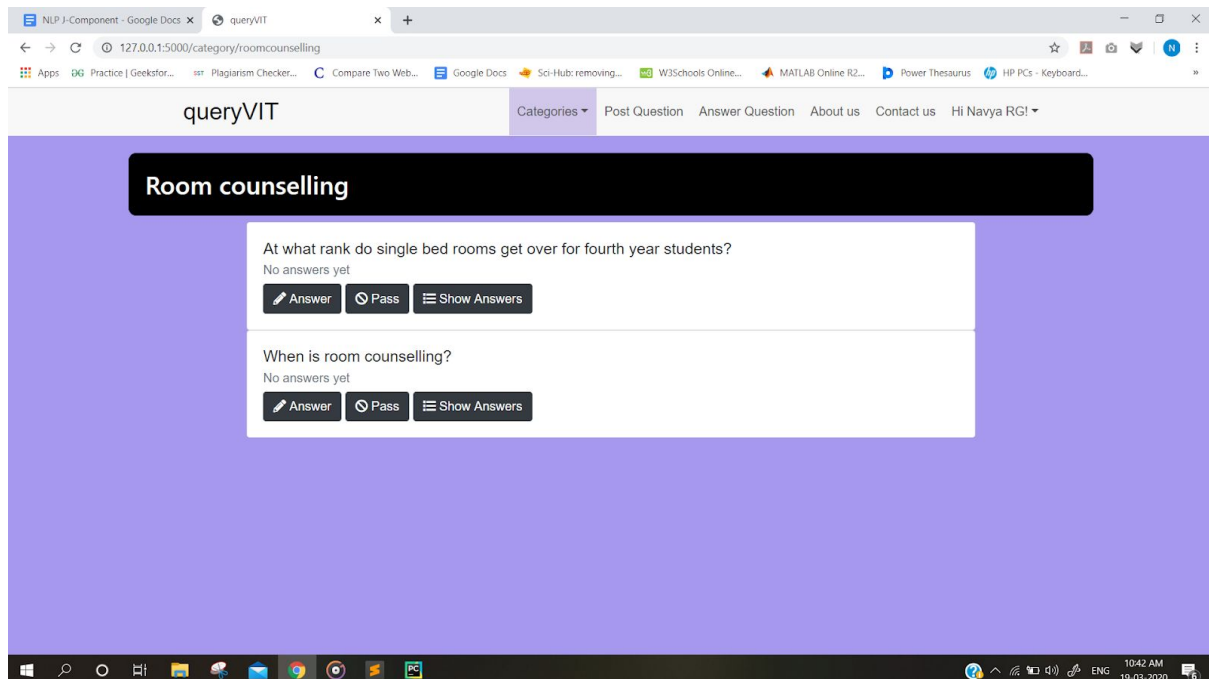
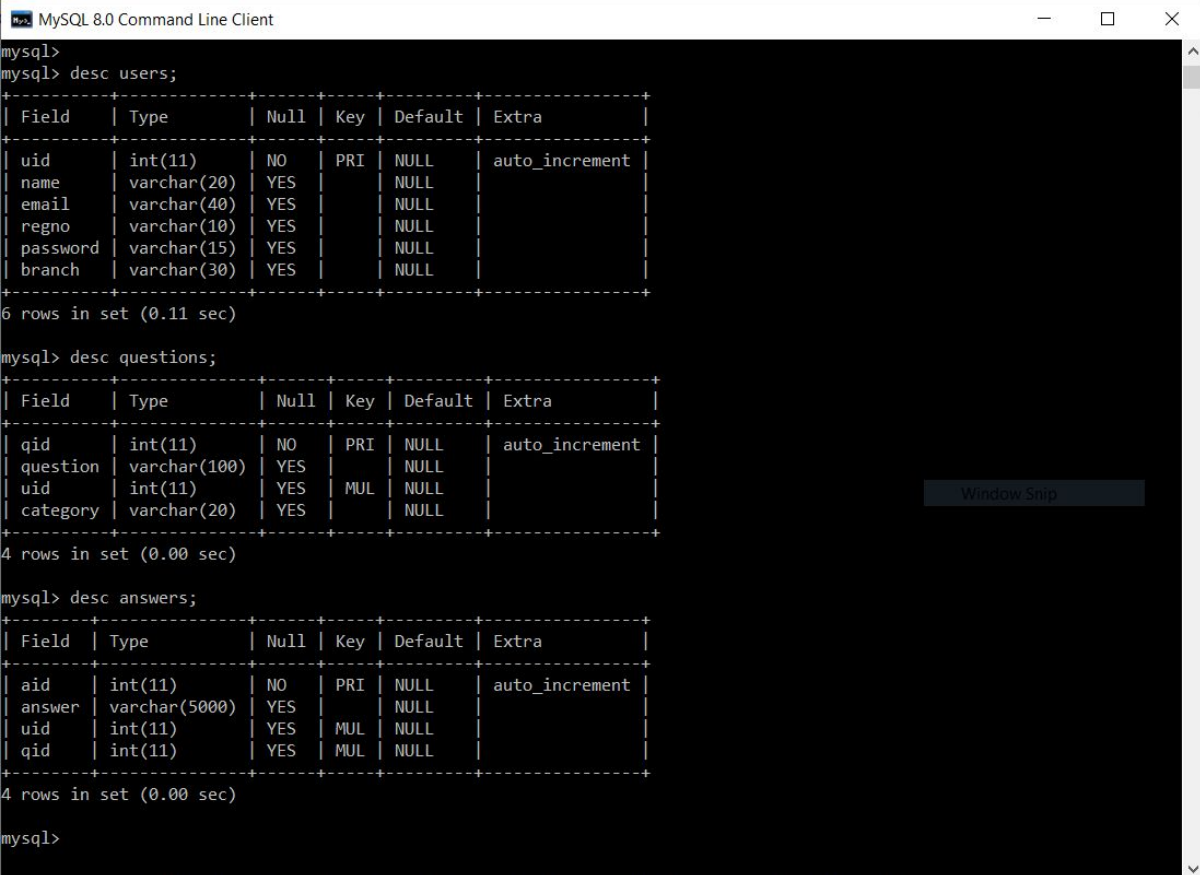


Fig 5.12: Category page

In Fig 5.12, the question entered was “When is room counselling?” in the category “room counselling”. Since there are no questions existing with a similarity score greater than 90%, the user is not redirected and the question is directly posted. After posting the question, the “room counselling” category web page is displayed and the user’s question can be found there.



```

mysql> desc users;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| uid   | int(11) | NO | PRI | NULL | auto_increment |
| name  | varchar(20) | YES | | NULL | |
| email | varchar(40) | YES | | NULL | |
| regno | varchar(10) | YES | | NULL | |
| password | varchar(15) | YES | | NULL | |
| branch | varchar(30) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.11 sec)

mysql> desc questions;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| qid   | int(11) | NO | PRI | NULL | auto_increment |
| question | varchar(100) | YES | | NULL | |
| uid   | int(11) | YES | MUL | NULL | |
| category | varchar(20) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> desc answers;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| aid   | int(11) | NO | PRI | NULL | auto_increment |
| answer | varchar(5000) | YES | | NULL | |
| uid   | int(11) | YES | MUL | NULL | |
| qid   | int(11) | YES | MUL | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>

```

Fig 5.13: Database schema

VIII. CONCLUSION

In the need for finding the most accurate answers to our queries, the model chosen finds the similarity between questions and provides a useful way to classify them. This allows new questions to be asked by ensuring that there is no repetition of questions, and making it easy for users to search for any query and find the most accurate question matching their query. The internet is flooded with information written by various authors and oftentimes, it can become tedious as there are hundreds of articles on the same topic and users can be unsure of what to believe. A simple solution to this is gathering all the similar information together and preventing redundancy. Our online forum succeeded in doing the same by not letting users ask questions which have already been asked and answered.

IX. REFERENCES

1. Takano, Y., Iijima, Y., Kobayashi, K., Sakuta, H., Sakaji, H., Kohana, M. and Kobayashi, A., 2019, March. Improving Document Similarity Calculation Using Cosine-Similarity Graphs. In *International Conference on Advanced Information Networking and Applications* (pp. 512-522). Springer, Cham.
2. Minaee, S. and Liu, Z., 2017, November. Automatic question-answering using a deep similarity neural network. In *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)* (pp. 923-927). IEEE.
3. Alodadi, M. and Janeja, V.P., 2015, October. Similarity in patient support forums using tf-idf and cosine similarity metrics. In *2015 International Conference on Healthcare Informatics* (pp. 521-522). IEEE.
4. Kanjirathinkal, R.C., Singh, A., Gangadharaiyah, R., Raghu, D. and Visweswariah, K., 2012, December. Does similarity matter? The case of answer extraction from technical discussion forums. In *Proceedings of COLING 2012: Posters* (pp. 175-184).
5. Suta, P., Mongkolnam, P., Fung, C.C. and Chan, J.H., 2018, December. Matching question and answer using similarity: An experiment with stack overflow. In *2018 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)* (pp. 51-54). IEEE.
6. Tóth, L., Nagy, B., Janthó, D., Vidács, L. and Gyimóthy, T., 2019, January. Towards an Accurate Prediction of the Question Quality on Stack Overflow Using a Deep-Learning-Based NLP Approach. In *14th International Conference on Software Technologies, ICSOFT 2019* (pp. 631-639). SciTePress.