



Summer Fellowship Report

On

Spam Filter Algorithm in Spoken Tutorial Forum

Submitted by

Pratik Ratadiya

Under the guidance of

Prof.Kannan M. Moudgalya

Chemical Engineering Department

IIT Bombay

July 7, 2018

Acknowledgment

I, the FOSSEE intern of the Spoken-Tutorial Forums Spam Filter Module, is overwhelmed in all humbleness and gratefulness to acknowledge my deep gratitude to all those who have helped me put my ideas to perfection and have assigned tasks, well above the level of simplicity and into something concrete and unique.

I, wholeheartedly thank **Ms. Nancy Varkey, Project Manager, Spoken-Tutorial.org** for having faith in me, selecting me to be a part of this valuable project and for constantly motivating the entire team to do better.

I am very thankful to my mentors **Mr. Saurabh Adhikary** and **Ms. Kirti Ambre** for their valuable suggestions. They were and are always there to show the right track when needed help. With help of their brilliant guidance and encouragement, I was able to complete my tasks properly and was up to the mark in all the tasks assigned. During the process, I got a chance to see the stronger side of my technical and non-technical aspects and also strengthen my concepts. Hereby, I feel myself honoured to have got a chance to work at this prestigious institute.

Last but not the least, I wholeheartedly thank all my other colleagues working in different projects for helping me evolve better with their critical advice.

With Regards.

Pratik Ratadiya
(Pune Institute of Computer Technology, Pune)

Contents

1 Abstract	3
2 Spam filter in spoken tutorials forum	5
3 Problems Encountered	7
3.1 Problem 1	7
3.2 Problem 2	8
3.3 Problem 3	9
3.4 Problem 4	10
3.5 Problem 5	11
3.6 Problem 6	11
4 Design considerations	12
5 Installation and updation guide	14
5.1 Installation	14
5.2 Updation	14
6 Code and working	16
7 Test cases and results	25
8 References	31

Chapter 1

Abstract

The Spoken Tutorial project is the initiative of the Talk to a teacher activity launched by the Ministry of Human Resources and Development, Government of India. It is being developed by IIT Bombay and provides spoken tutorials on FOSS available in several Indian languages, for the learner to be able to learn in any language he/she is comfortable in.

For providing assistance to the learners, teacher guardians, and contributors, a forums website (forums.spoken-tutorial.org) has been set up. Users can ask their queries over here which are then cleared by the tutorial creators, website administrator and other users as well. The intention of the forums is to provide a platform between the users and creators, administration and lead to better working and development of the community.

However as in the case with existing forums websites, there have been instances of some users posting malicious content, spam links on the website. This leads to digression from the core objectives of the site and also possesses harm to the user profiles as well as the website. Also, teachers and students tend to ask questions not related to the videos under the category of tutorials which then appear on the questions panel of the respective tutorial causing unnecessary distraction. Thus proper handling of such cases has become a necessity.

Manual moderation can be used for this purpose but it increases the cost as well as dependency for completion of the objective. We intend to build an automated spam filter which does the job automatically with acceptable accuracy and properly guides the users as well. The program will make use of a training dataset and use machine learning algorithms to satisfy the purpose.

Chapter 2

Spam filter in spoken tutorials forum

The spam filter has been created to able to successfully filter out content entered by user which is either malicious, spam or has been posted at irrelevant place. It does this with the use of machine learning algorithms and an already labelled dataset. There are users who use the forums website for the publicity of their products, websites etc. Teachers and students tend to post questions related to conducting exams, certificates, login difficulties etc. directly under a video. The current system is designed such that video related to topic runs on the left side while questions asked for that topic appear on the right side of the video. As a result, these irrelevant questions are visible to the viewer on playing the video. Spammers also introduces malicious links under a thread which digresses the discussion. To avoid this a spam filter system needed to be developed. At present, this work was done by manual moderation. However this increases the external dependencies for execution. As a result, an automated procedure will be more beneficial as it filters the comment before it gets uploaded itself. With the available dataset of previously asked questions, replies and answers we train a model on how exactly does a spam comment look like. Once trained, it is then able to predict the nature of

entered comment. There will be a 2 layered filtering comprising of:

1. Whether the entered content is spam or not
2. If it is not a spam, then check whether it is a tutorial related question or a general question (related to payment, certificates, conduction of tests etc)

Tutorial related questions are to be accepted by the system. In case of a spam question is asked, it should be obstructed from being posted. However, the user has an option of sending such a question for admin review, upon which appropriate action will be taken. If a general question is asked under a different category, user should be asked whether he wishes to post it under the general category forum and accordingly the question will be accepted or discarded.

Chapter 3

Problems Encountered

There were a number of problems faced while creating this module. After technical analysis and discussion with mentors, the most feasible solutions were chosen and worked upon. The following were the problems encountered in the creation of spam filter module:

3.1 Problem 1

Inadequate and unlabelled data: Majority of the spam instances encountered in the past had been deleted. As a result there was inadequate data. Also, there was no labelling of the present dataset. As spam filtering module is made through a supervised learning procedure, labelled as well as adequate data was necessary.

Solution:

1. For inadequate data: Previously encountered questions and queries were collected from different contributors working at Spoken Tutorials. The data was then stored in a comma separated values(csv) file. A total of 406 samples were collected. Further, samples were increased synthetically using certain techniques(discussed later).

2. For unlabelled data: Samples were labelled manually over a course of time. Previously stored questions having their status bit 0 in the database were also added directly to the database.

3.2 Problem 2

Distribution of data and labelling based on two layer filtering: As a two way filter was to be implemented, two different labels would be attached to every data sample for training purposes. The file structure for storing data in accordance with this two way filtering was also to be determined.

Solution:

The initially collected data was stored in a file(Stdataset.csv). A subset of the data was stored into another csv file(OnlySpam.csv) which was to be used for the first layer of filtering(Spam vs not spam). The file composition of each of these files is as follows:

1. Stdataset.csv: File containing two columns: Content and Label. There are 406 samples in this file. Content column contains the text data. Label is a binary value with 0 being related to tutorial videos and 1 indicating content being not related to tutorial. Out of the 406 samples, 281 samples were related to the tutorial, while 125 samples were not related to the tutorial.(30.78%)
2. OnlySpam.csv: A subset of data from Stdataset.csv, along with some commonly found spam statements on the internet were added to form this file. It has columns identical to the first file. However, here 0 bit in Label column indicates sample being not spam while 1 bit indicates the

sample being spam. Out of the 318 samples, 274 were not spam while 44 were spam samples.(13.83%)

3.3 Problem 3

Pre-processing of the training data: The data which has been stored into two files now needs to be trained. However, it cannot be used effectively in the current state and needs to be filtered and processed for better results.

Solution:

We pass the data through certain filtering as:

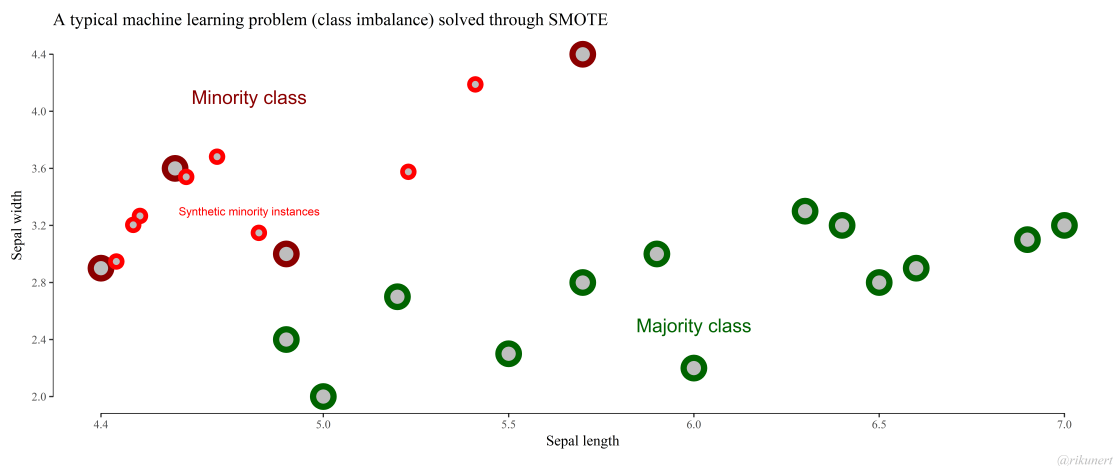
1. Stripping of HTML tags: The text is removed of any html tags and scripts as they are not relevant to the intent of the message. Further, removal of scripts reduces the risk of any cyber attack on the website. Apart from this special characters like — ,newline and tab characters are also removed. Further words with less than 3 characters are removed as they do not provide relevant information and tend to divert the model function.
2. Stop words: Commonly used words like 'are', 'is', 'they', 'this' etc can be found in both spam as well as non-spam content and thus cannot be a deciding factor. Such words are called stop words and we remove them from our text.
3. Vectorization: The filtered text is now converted into a sparse matrix of tokens by vectorizing them. Tf-Idf(Term frequency inverse document frequency) vectorizer is used for this purpose.

3.4 Problem 4

Scarcity of minority class: As noted earlier, the percentage of minority class in both datasets was less than 35%. Hence, the trained model will tend to have a bias for the majority samples, thus leading to a decrease in the precision of the module.

Solution:

We use a technique known as Synthetic Minority Over-sampling Technique (SMOTE) for synthetically increasing the number of samples labelled 1.



In this technique for each one of the n neighboring minority cases we have randomly chosen we add a synthetic case somewhere between that neighboring minority case and the original minority case. In this last sentence between means within the exact straight line that passes through the 2 cases, and somewhere means the synthetic positive case is randomly placed in between.

3.5 Problem 5

Selection of supervised machine learning model:

For training the processed data, we need a suitable supervised machine learning algorithm which provides the best predictor function

Solution:

We tested various algorithms over the datasets and it was found that the Linear SVC (Support Vector Classifier) gave the best results. It is a linear kernel implementation of the SVM algorithm which creates a boundary between the two classes for classification purpose.

3.6 Problem 6

Reducing required runtime for the model: Every-time the page is loaded, calling the entire script leads to increase in the load time of the program which leads to a bad experience for the user.

Solution:

We separate the model file and the predictor model file. The trained model file is saved in .sav format. When a user enters any text, the text is cleansed by the above mentioned method and then sent to the model for prediction. Thus, the run time is only equal to the runtime of the predictor script.

Chapter 4

Design considerations

The following libraries and packages were used in the creation of the module:

Pandas: Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. We use it to create dataframe from the csv files. Dataframes simplify the processing of text.

Beautiful Soup: Beautiful Soup is a Python package for parsing HTML and XML documents. It creates a parse tree for parsed pages that can be used to extract data from HTML, which is useful for web scraping. We use it to strip off HTML tags form our entered text.

Sklearn: Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms as well as other useful tools required for constructing a machine learning model. We import this library to use vectorizers and also the LinearSVC module.

Pickle, Re, lxml and imblearn:

1. Pickle: It is used to save and load the model file stored in .sav format.
2. lxml: Lexical XML parser used by beautiful soup for parsing the text.
3. Re: Re is a python library which is used to compile, substitute and work with regex expressions
4. Imblearn: We import this library to use SMOTE package used for synthetic minority oversampling.

Note that any internal dependencies required for the above mentioned packages need to be installed in your environment as well

Chapter 5

Installation and updation guide

5.1 Installation

For the complete functionality of the module, the fore-mentioned packages as well as some other libraries need to be installed. The prominent ones of these are as follows:

Package	Version
Pandas	0.22.0
lxml	4.2.1
Numpy	1.14.3
Scipy	1.1.0
Scikit-learn	0.19.1

These as well as other requirements are stored in the mlrequirements.txt file present in the main directory. Install them using pip by running the command

```
pip install -r mlrequirements.txt
```

The system is now ready to be used.

5.2 Updation

To make any changes in the model or in the dataset in the future, perform the following steps:

1. Add/edit/delete data sample to the files Stdataset.csv and OnlySpam.csv present in the app directory i.e. alongside manage.py file of the website. Make sure that data is added in the correct format and label. Stdataset.csv file is used to train tutorial(0) vs training(1) model while OnlySpam.csv file is used for spam(1) vs not spam(0) model.
2. Rerun the file Spoken.py or OnlySpam.py depending on if you have made changes in Stdataset.csv or OnlySpam.csv respectively. The commands for the same are:

```
python Spoken.py  
python OnlySpam.py
```

On execution of the script, restart the server and the spam module will be updated. Also, make sure that the added samples don't lead to excessive gap between majority class and the minority class.

Chapter 6

Code and working

Following are the codes written for performing various functions and they are defined as follows:

1. Training model file:

```
    # Libraries
import pandas as pd
from bs4 import BeautifulSoup
from sklearn.feature_extraction.text import
    TfidfVectorizer
from sklearn.svm import LinearSVC
import re
from nltk import word_tokenize
from nltk.stem import WordNetLemmatizer
import pickle
from imblearn.over_sampling import SMOTE
fields = ['Content', 'Label']

#Load into dataframe
df =
    pd.read_csv('STdataset.csv', skipinitialspace=True, usecols=fields)

#Stripping function
def remove_tags(text):
    soup = BeautifulSoup(text, "lxml")
```

```

    if soup.find_all('style'):
        soup.style.decompose()
    string = soup.get_text()
    string =
        string.replace('&nbsp;','').replace('\n','').replace('\r','')
    string = ' '.join([w for w in string.split() if
        len(w)>=3])
    return string

df['Content']=df['Content'].apply(remove_tags)

#Lemmatizer
class LemmaTokenizer(object):
    def __init__(self):
        self.wnl = WordNetLemmatizer()
    def __call__(self, doc):
        return [self.wnl.lemmatize(t) for t in
            word_tokenize(doc)]

#Vectorizer
vectorizer =
    TfidfVectorizer(stop_words='english',tokenizer=LemmaTokenizer())

x = vectorizer.fit_transform(df['Content'])

#Minority oversampling
sm = SMOTE(random_state=42)

x,y = sm.fit_sample(x,df['Label'])

#Model fitting
model = LinearSVC(random_state=42,
    tol=5,fit_intercept=False)
model.fit(x,y)
filename = 'tutorial_model.sav'
pickle.dump(model, open(filename, 'wb'))

```

2. Prediction function:

```
#Predictor function
def predictor(comment):
    simplified = remove_tags(comment)
    tester = [simplified]
    print(simplified)
    contest = vectorizer.transform(tester)
    load_model = pickle.load(open(filename, 'rb'))
    a = load_model.predict(contest)
    return a[0]
```

3. Views in views.py:

```
#For new tutorial questions
def new_question(request):
    context = {}
    if request.method == 'POST':
        if request.POST['action'] == 'Submit
Question':          # Submitted first time
            content = request.POST['body']
            title = request.POST['title']
            resultspam = predictorsspam(content)
                        # Check for spam or not
            spam
            warning = ''
            if resultspam == 1:          # If spam
                return back with option for review
            warning = 'Our system detects you have
                entered a possibly spam content. Do
                you want admin to review the same?'
            context['help'] = warning
            category =
                request.POST.get('category', None)
            tutorial =
                request.POST.get('tutorial', None)
```

```

context['tut'] = tutorial
minute_range =
    request.POST.get('minute_range',
        None)
context['minute_range']=minute_range
second_range =
    request.POST.get('second_range',
        None)
context['second_range']=second_range
# pass minute_range and second_range
# value to NewQuestionForm to
# populate on select
form =
    NewQuestionForm(category=category,
        tutorial=tutorial,
                                minute_range=minute_range,
                                second_range=second_range)
soup = BeautifulSoup(content,"lxml")
if soup.find_all('style'):
    soup.style.decompose()
content = soup.get_text()
context['body'] =
    content.lstrip().rstrip()
context['title2'] = title
context['form'] = form
return render(request,
    'website/templates/new-question.html',
    context)

```

```

resultpredictor = predictor(content)    #
    If not spam check if related to tutorial
if resultpredictor == 1:                #
    If not related to tutorial return back
    with option for general
    warning= 'Our system detects you have
        possibly entered a general
        question. Do you want to post it

```

```

        over there?'
    category =
        request.POST.get('category', None)
    tutorial = "General"
    minute_range = None
    second_range = None
    context['help']=warning
    context['category']=category
    # pass minute_range and second_range
    # value to NewQuestionForm to
    # populate on select
    form =
        NewQuestionForm(category=category,
            tutorial=tutorial,
                                minute_range=minute_range,
                                second_range=second_range)
    soup = BeautifulSoup(content, "lxml")
    if soup.find_all('style'):
        soup.style.decompose()
    content = soup.get_text()
    context['body'] =
        content.lstrip().rstrip()
    context['title'] = title
    context['form'] = form
    return render(request,
        'website/templates/new-question.html',
        context)

form = NewQuestionForm(request.POST) #
    else form new instance and assign
    parameters
if form.is_valid():
    cleaned_data = form.cleaned_data
    question = Question()
    question.uid = request.user.id
    question.category =
        cleaned_data['category'].replace(' ',

```

```

        '-')
question.tutorial =
    cleaned_data['tutorial'].replace(' ',
        '-')
question.minute_range =
    cleaned_data['minute_range']
question.second_range =
    cleaned_data['second_range']
question.title = cleaned_data['title']
question.body =
    cleaned_data['body'].encode('unicode_escape')
if request.POST['action'] == 'Send for
review': # if question submitted for
review hide it by status
    question.status = 0
    question.views = 1
    question.save()
    messages.success(request, "Your
        question has been sent for review.
        Check the site for further
        updates!")
    return HttpResponseRedirect('/')
question.views = 1

question.save()
return HttpResponseRedirect('/')
else:
    # get values from URL.
    category = request.GET.get('category', None)
    tutorial = request.GET.get('tutorial', None)
    minute_range =
        request.GET.get('minute_range', None)
    second_range =
        request.GET.get('second_range', None)
    # pass minute_range and second_range value to
    NewQuestionForm to populate on select
    form = NewQuestionForm(category=category,

```

```

        tutorial=tutorial,
                                minute_range=minute_range,
                                second_range=second_range)
    context['category'] = category

    context['form'] = form
    context.update(csrf(request))
    return render(request,
                  'website/templates/new-question.html', context)

```

```

#For training related questions
def new_question_general(request):
    context = {}
    if request.method == 'POST':
        if request.POST['action'] == 'Submit
        Question':
            content = request.POST['body']
            title = request.POST['title']
            resultspam = predictors.spam(content)
            warning = ''
            if resultspam == 1:                # If spam
                return back with option to review
            warning = 'Our system detects you have
                entered a possibly spam content. Do
                you want admin to review the same?'
            context['help'] = warning
            category =
                request.POST.get('category', None)
            tutorial =
                request.POST.get('tutorial', None)
            minute_range =
                request.POST.get('minute_range',
                None)
            second_range =
                request.POST.get('second_range',
                None)
            # pass minute_range and second_range

```

```

        value to NewQuestionForm to
        populate on select
    form =
        NewQuestionForm(category=category,
            tutorial=tutorial,
                                minute_range=minute_range,
                                second_range=second_range)
    soup = BeautifulSoup(content, "lxml")
    if soup.find_all('style'):
        soup.style.decompose()
    content = soup.get_text()
    context['body'] =
        content.lstrip().rstrip()
    context['title'] = title
    context['form'] = form
    return render(request,
        'website/templates/new-question-general.html',
        context)

form = NewQuestionForm(request.POST)    #
    Else accept
if form.is_valid():
    cleaned_data = form.cleaned_data
    question = Question()
    question.uid = request.user.id
    question.category =
        cleaned_data['category'].replace(' ',
            '-')
    question.tutorial =
        cleaned_data['tutorial'].replace(' ',
            '-')
    question.minute_range =
        cleaned_data['minute_range']
    question.second_range =
        cleaned_data['second_range']
    question.title = cleaned_data['title']
    question.body =

```



```

        cleaned_data['body'].encode('unicode_escape')
    if request.POST['action'] == 'Send for
    review': # If sent for review hide it
        question.status = 0
        question.views = 1
        question.save()
        messages.success(request, "Your
        question has been sent for review.
        Check the site for further
        updates!")
        return HttpResponseRedirect('/')
    question.views = 1
    question.save()

    return HttpResponseRedirect('/')
else:
    # get values from URL.
    category = request.GET.get('category', None)
    tutorial = request.GET.get('tutorial', None)
    minute_range =
        request.GET.get('minute_range', None)
    second_range =
        request.GET.get('second_range', None)
    # pass minute_range and second_range value to
    NewQuestionForm to populate on select
    form = NewQuestionForm(category=category,
        tutorial=tutorial,
                                minute_range=minute_range,
                                second_range=second_range)

    context['category'] = category
    context['form'] = form
    context.update(csrf(request))
    return render(request,
        'website/templates/new-question-general.html',
        context)

```

Chapter 7

Test cases and results

S No.	Test Case	Expected Result	Actual Result
1	User entered a tutorial related question in tutorials related questions page eg:I have a doubt regarding initialisation of variables in Biopython. There is an error	The question gets accepted and user is redirected to the home page	The question gets accepted and user is redirected to the home page

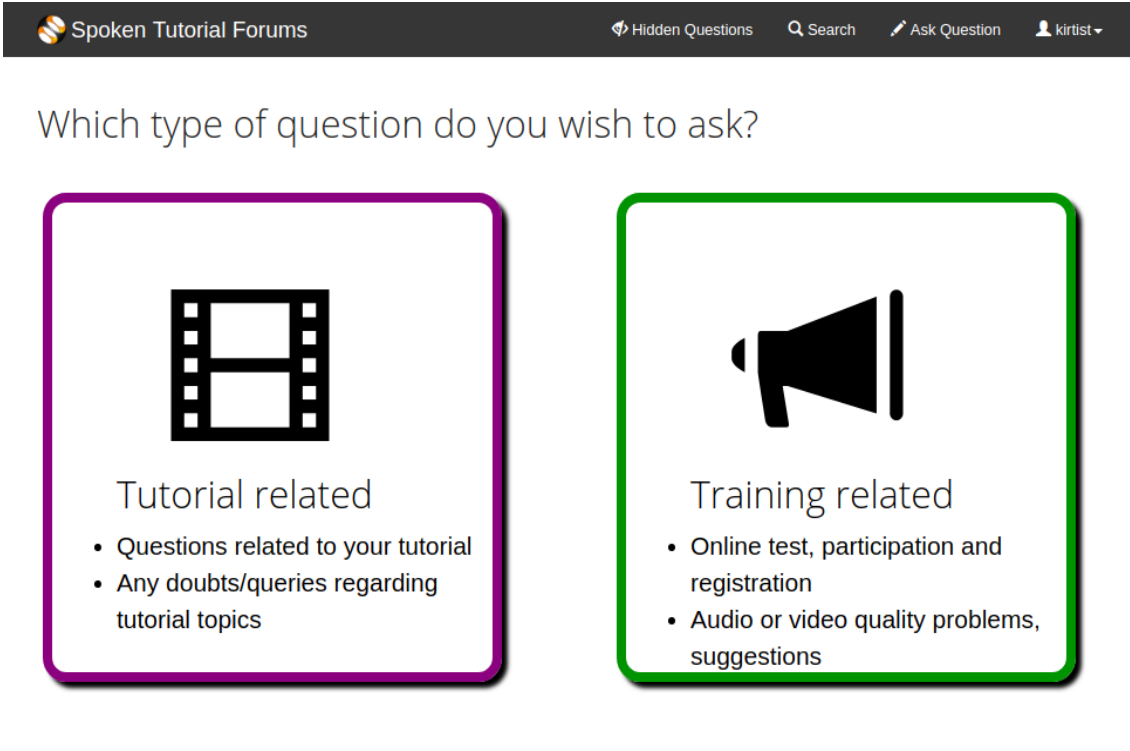
S No.	Test Case	Expected Result	Actual Result
2	User entered a training related question in tutorials related questions page eg:How can we take a certificate test for this course??	A popup appears stating that question is a general question and does the user wish to post it over there. If yes, question is saved else discarded and user is redirected to home	A popup appears stating that question is a general question and does the user wish to post it over there. If yes, question is saved else discarded and user is redirected to home
3	User entered a spam question in tutorials related questions page eg:Industrial training in latest technologies and placement guarantee. www.timesjobs.com or call 7689029011	A popup appears stating that question is a spam question and does the user wish to send the question to admin for review. If yes, question is sent for review else discarded and user is redirected to home	A popup appears stating that question is a spam question and does the user wish to send the question to admin for review. If yes, question is sent for review else discarded and user is redirected to home

S No.	Test Case	Expected Result	Actual Result
4	<p>User entered a spam question in training related questions page eg:Apply for pan card online</p>	<p>A popup appears stating that question is a spam question and does the user wish to send the question to admin for review. If yes, question is sent for review else discarded and user is redirected to home</p>	<p>A popup appears stating that question is a spam question and does the user wish to send the question to admin for review. If yes, question is sent for review else discarded and user is redirected to home</p>
5	<p>User entered a non spam question in training related questions page eg:For more information regarding inheritance and polymorphism, which websites should I refer to?</p>	<p>The question gets accepted and user is redirected to the home page</p>	<p>The question gets accepted and user is redirected to the home page</p>

Results

For a test set of 40 samples collected from the spoken tutorials team, the model was able to successfully classify 36 samples with the desired output. The following are the screenshots of the corresponding outputs shown on the screen for various test cases:


1. Question category page



Spoken Tutorial Forums


Hidden Questions Search Ask Question kirtist

Which type of question do you wish to ask?



Tutorial related


- Questions related to your tutorial
- Any doubts/queries regarding tutorial topics




Training related

- Online test, participation and registration
- Audio or video quality problems, suggestions

2. Training related question in tutorials category

 Spoken Tutorial Forums [Hidden Questions](#) [Search](#) [Ask Question](#) [kirtist](#)

 Create a new question . . .




Please enter the tutorial details.

Please enter your question details.

Our system detects you have possibly entered a general question. Do you want to post it over there?


Title:


Question:

Font Size... **B** *I* U ~~X~~ ^{X²}   

I had applied for certificate in Java after completing all tutorials but haven't received the same. Whom shall I contact?

3. Spam related question in tutorials category

 Spoken Tutorial Forums [Hidden Questions](#) [Search](#) [Ask Question](#) [kirtist](#)

 Create a new question . . .




Please enter the tutorial details.

Please enter your question details.

Our system detects you have entered a possibly spam content. Do you want admin to review the same?

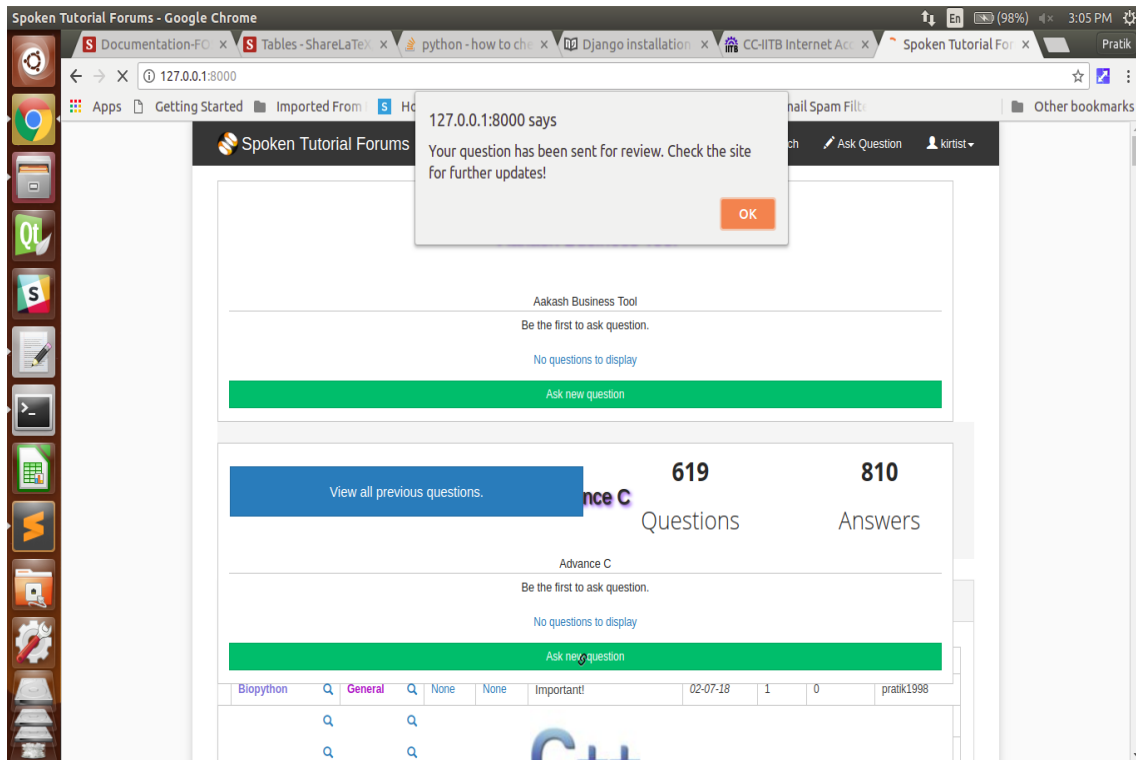
Title:

Question:

Font Size... **B** *I* U ~~X~~ ^{X²}   

Industrial training in latest technologies and placement guarantee. www.timesjobs.com or call 7689029011

4. Spam question sent for review by user



Thus we have been able to produce satisfactory results through the use of this module.

Chapter 8

References

1. Pandas:
<https://pandas.pydata.org>
2. Scikit learn:
<https://scikit-learn.org>
3. SMOTE:
Research paper on SMOTE here
4. Linear SVC:
Official documentation link here