ABSTRACT

The Evaluation of answer sheets is an important part of tracking academic performance in students in schools and colleges and other educational institutions. Usually, the method used to evaluate answer sheets is by a human teacher which can be biased. It can be affected by various factors such as relation between exam giver and exam checker and the external conditions in which the evaluator is present. Moreover, manual evaluation of answer sheets can be an onerous and laborious task. The concept of natural language processing plays an important role in the process of automatic answer sheet evaluation.
NLP or Natural Language Processing can be defined as a way of interaction between humans and computers. It is a field which attracts the interest of a lot of machine learning enthusiasts due to its ability to statistically read natural language despite it being subjective.

Machine Learning does not require humans to manually drive them. Instead the system takes in the data and stores it for its own knowledge. It then learns from this data to make accurate decisions. Machine Learning has helped us in designing our own system for Automatic Descriptive Answer Evaluation and is also a very important part of the project.

Various methods have been implemented for Automatic Answer Evaluation of Subjective or Objective answers but the methodologies used by us have helped us in the evaluation of descriptive answers. Since there is a plethora of data available, developments in systems are extremely important to keep the knowledge base and the user base more efficient.

Here we present a report on how people, especially teachers can use Automatic Descriptive Answer Evaluation for the checking of examination sheets. The bare minimum human intervention makes this a very effective and useful model in personal as well as professional life.

Keywords : Machine Learning, Automatic Descriptive Answer Evaluation, weighted parameter value, similarity measure, Text Summarization, Natural Language Processing.

1. INTRODUCTION

1.1 Overview

There are various assessment strategies used to assess a student's performance. The most used technique is the response to a descriptive question. In this technique, a student expresses their opinion in response to the question verbatim. The descriptive response automatic rating system will be very cooperative for various universities and educational institutions to very effectively assess a student's performance.

A student can answer a question using different grammatical styles and choosing different words that are similar to the actual answer. The motivation behind the automated response script assessment comes from less time, less workforce involvement, prohibition of psychological changes in the human assessor, and is very easy to keep and retrieve. It also ensures that mood swings or changes in perspective of the human reviewer will not affect the review process. The automatic evaluation of the response script will help us overcome the difficulties encountered in the manual evaluation. Here, a student's written response is provided as input and the system will automatically score grades after the assessment. The system takes into account all possible factors such as spelling error, grammatical error, and cosine similarity measures for scoring marks.

1.2 Objective

In today’s world, with unlimited storage and everything being internet based, it is important to make models with features that reduce human intervention and give maximum efficiency, making lives easier for everyone who requires the use of technology. Agile steps must be taken to make advanced and intricately designed techniques which help the people with their personal and professional lives. This project aims at reducing human intervention while evaluating answer sheets of students and making the whole process automatic so as to get rid of any bias that may arise due to manual evaluation and also to make the task faster and more efficient.

1.3 Scope

The scope of this project extends to both personal and professional uses. It would come in handy for a variety of applicants like :

<u>Answer Sheet Evaluation</u>: (Digital Correction of Exam Papers)

There is no location constraint in such a case. Any examiner/ answer sheet checker or moderator can verify/evaluate answer sheets sitting at their location.Person can securely log in to the system and can evaluate scanned answer sheets. The technology-driven process is efficient and the time needed to evaluate answer sheet reduces significantly as physical handling of answer sheet is eliminated. Logistical cost and travel management of each evaluator/moderator is eliminated. The system helps to auto calculate the total. As an examiner you need not have to keep track of compulsory, optional questions attempted by the students. The system takes into consideration best of performances of the students for optional answers.The calculation of the total is as per the question paper pattern and marking scheme. It eliminates manual processes. As an examiner you can save 5 to 10 minutes of total calculation activity.

<u>Result Generation</u>

It becomes automated as the system can directly calculate the result and can generate mark sheets instantly. It can eliminate the process of manually entering marks in the software.There is a facility to export the result in excel format. This format can be imported in any of the result generation solutions. Manual data entry work of entering marks, validation of it can be eliminated.It would help to speed up the result generation process.It is mandatory to declare results for the university within 45 days of examination dates. There is a timeline and pressure to complete the entire answer sheet evaluation activity within the stipulated timeline. An onscreen evaluation system can help to simplify the result generation process. The system is defined in such an away that calculation of total marks obtained by the student is auto based on the exam pattern.If the student has attempted 4 questions out of 5 and there is the instruction of attempt any 3 out of 5 then the system would consider the best of 3 scores while calculating the results.

Examiners need not have to keep track of it.It saves time on manual score calculations. The entire result of the individual as well as a group of students is available on a single click button and it can be exported in the excel sheet as well.

Student request for scan copy

It can be managed easily from the software system .As per rules and regulations, most of the institutions and universities allow students to see their evaluated answer sheet copies. Students need to make payments to get access to their evaluated answer sheets .In case of such requests, the institution needs to identify physical answer sheet copy, prepare a photocopy of it and issue it to the respective students. There is a significant logistical and administrative activity in this process. An onscreen evaluation system can eliminate all administrative hassles and can simplify this process. Students can easily see digitally evaluated answer sheets using technology. If particular student requests for such a copy then the administrator can assign View access right to such student so that students can see individual answer sheet copy online itself.

Re-Evaluation of Answer Sheet :

There are cases when the students are not satisfied with the results, and request for revaluation or rechecking of answer sheets. Onscreen evaluation makes it simpler to do this activity. This process can be completed in record time. It can help to generate results in a quick time.

Logistical Management:

Logistical management is the organization of the question papers and answers sheets before the examination, after the examination and during the paper checking as well. Physically answer sheet to be stored in a central location insecure environment. The exam paper checker needs to visit this centralized location in order to evaluate the answer sheets of the individual students. Location is one of the important constraints in such a situation. If there are thousands of answer sheets to be evaluated then you may need hundreds of evaluators who should visit this central location and evaluate answer sheets in a secure manner. Moderators are also expected to visit the location to moderate / recheck answer sheets. The time consumed for this activity along with the cost is higher. In some cases, universities or institutes prefer to send answer sheet copies through courier at examiner location. It involves delays and coordination activities. Physical handling of the answer sheet added to delays of result processing. This integrated manner of dealing with the handling of the answer sheets is risky as well as extremely time-consuming.

Identity Disclosure:

Whilst any examination process, the privacy or say the identity of the candidate has to be secured. It is one of the most mandatory requirements of examination conduction. In the traditional way of answer sheet evaluation, it is essential to hide student details to avoid malpractices. Manually each answer sheet should be arranged so as to hide the identity of the student. This makes the identification of the candidate more obvious and the safety regarding the privacy of the student was much more at risk. Apparently, there was a need to take action for the reduction of the errors that take place during the earlier methods of examination evaluation. The perfect and most needed solution to the same was the adoption of the Onscreen Evaluation System, which is designed especially for the purpose of paper checking.

1.4 Outline

This project report consists of the following chapters and References. Rest of the report is organized as following:

Chapter 2

This chapter deals with the analysis of the problem statement. The literature survey has been done so as to chalk out the advantages and limitations and draw conclusions from the same. It also represents a few case studies of pre-existing Automatic Answer evaluation algorithms.

Chapter 3

This chapter depicts all the requirements, both hardware and software, required to complete our project. It highlights the basic design and architecture of our system. It gives a design approach and design methodology of our model. It gives the whole step by step implementation of the project with the help of code snippets, flowcharts and images. It gives the workflow and the procedure part of the project. It takes into account the users, user queries, requirements, microservices used and created to serve the functionalities of the project.

Chapter 4

This chapter gives us the results concluded from our project and the step by step analysis of each phase of the project. It also highlights the challenges we faced during the implementation of the whole project and how we overcame them.

Chapter 5

In this chapter we have stated the conclusion that we arrived at working on this project.

2. LITERATURE REVIEW

This chapter refers to the acute and precise information collected from various research papers and journals. The techniques and various advantages have been thoroughly in the chapter.

2.1 Problem Statement

Objective:

To develop an automated system for answer evaluations in online exams by using cosine similarity, fuzzy logic and Natural Language Processing concepts for the purpose of identifying the similarities in the student's answers and teacher's answers.

Relevance:

The solution should be up to an edge to the current existing comparable models so that the model has more accuracy and does the work in less time and produces a lesser amount of errors in the evaluation and summarization process.

Issue:

The issue that arises is that the process of summarizing the large text answers and then evaluating them can multiply the errors and to produce a machine that does the evaluation process with more accuracy and speed as compared to already existing answer evaluation models.

2.2 Analysis

SNo.	PUBLISHED IN	NAME OF RESEARCH PAPER	AUTHORS	BRIEF SYNOPSIS
1.	International Journal of Scientific Research and Engineering Development	Semantic Analysis and Evaluation of Subjective Passage	Jaco van de Pol and Michael Weber	An automatic answer checker application that checks and remarks written answers similar to human beings. The purpose of this system is to automate the old fashioned manual system and introduce automatic evaluation of marks in much faster and accurate way
2.	The University of Edinburgh Research and Teaching	WME and Automatic Mathematical Answer Checking	G Hanna, DA Reid & M de Villiers	Answer Checking The usefulness and challenges of automatic checking (grading/marking) of mathematical answers have been answered and provide a response page. The user input is displayed using MathML and transmitted to the answer checker in infix notation

3.	Journal of Technology Learning and Assessment	Semantic Analysis and Evaluation of Subjective Passage	Douglas Grimes & Mark Warchauer	Automated writing evaluation software that uses artificial intelligence to score student essays and support revision. Although many teachers and students considered automated scoring unreliable, and teachers' use of AWE was limited by the desire to use conventional writing methods, use of the software still brought important benefits.
4.	The University of Edinburgh Research and Teaching	WME and Automatic Mathematical Answer Checking	G Hanna, DA Reid & M de Villiers	The usefulness and challenges of automatic checking (grading/marking) of mathematical answers have been answered and provide a response page. The user input is displayed using MathML and transmitted to the answer checker in infix notation
5.	International Journal of Scientific Research and Engineering Development	Evaluating automatic detection of misspellings in German	Khanderao Biradar, Snehal Chaskar , Mansi Hirve , Rahil JaiswaL	Matching of misspellings with their respective target words, which is important in evaluating the spell checkers success in Evaluating Automatic Detection of Misspellings in German In the case of incorrect input, students choose to revise and resubmit their answer
6.	Institute of Information Science, Academia Sinica, Taiwan	An automatic multiple-choice question generation scheme for english adjective understanding	Yi-Chien Lin, Li-Chun Sung, Meng Chang Chen	some automatic methodologies were proposed . To avoid the first impediment, the question description in these methodologies is just the sentence containing the quizzing target modified by replacing the quizzing target Usage Checker Answer candidates
7.	12th Conference of the European Chapter of the ACL	Text-to-text Semantic Similarity for Automatic Short Answer Grading	Michael Mohler and Rada Mihalcea	Unsupervised techniques were used for evaluation of short answers.They use knowledge based and corpus based similarity, and effected by domain and size of corpus . They added automatic feedback from student answer in order to improve the performance..

				However,grammatical and spelling errors were not taken into account when evaluating

Table 2.1 . Analysis of Various Research Papers

3. PROPOSED METHODOLOGY

This chapter highlights the basic design and architecture of our system. It gives a design approach and design methodology of our model. It basically tells how we initiated the project and using which technologies. It also gives a brief preview of the architecture behind the full project. It focuses on the solution that we have proposed to eliminate the limitations encountered and the selection of different hardware and software tools required for the implementation of the Descriptive Answer Evaluation.

3.1 The Design

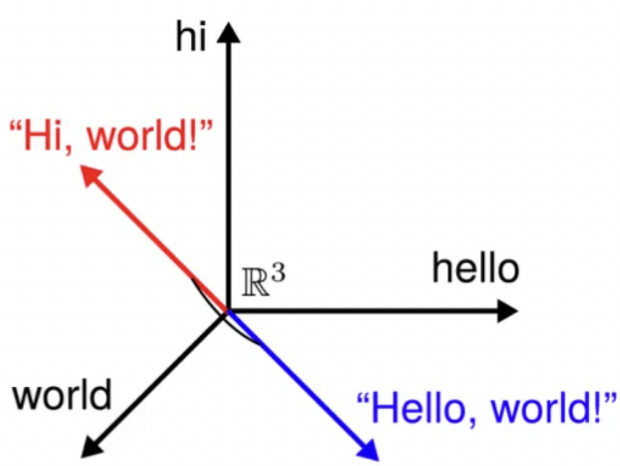

Fig. 3.1 A figure used to show the Cosine similarity between two sentences.

We usually use Cosine similarity for automatic evaluation of answers. The biggest advantage of this method is that it can be done regardless of the difference of the size of the two documents that are to be analysed. Evaluation of keywords is done based on the results of pre-processing. It is based on Cosine Similarity between the student's answer and the model answer provided by the teacher. The implementation involved here is the conversion of documents into vectors that are present in a multidimensional system. After this conversion the 'cos' of the angle between these two vectors is calculated. Lesser the angle, the more the similarity between the documents. This methodology helped us in faster answer evaluation and allotting of marks. We also used Fuzzy Logic and Naive Bayes algorithms to get better and more efficient results. So, the design of our project basically revolves around 4 main concepts - Natural Language Processing, Cosine Similarity, Fuzzy Logic and Naive Bayes algorithm.

3.2 Background

Dataset

For our model, our dataset was basically about 150-200 answer sheets that were sampled. We made an application on a cloud platform called Heroku for the students to fill in their answers. These answer sheets were then converted to a csv file. The csv file was then preprocessed for tokenizing the words present in it and for the implementation of stemming and lemmatization. This tokenized file was then sampled against the 'ideal' answer provided by us. Our provided answers were also tokenized for easier computation.

Basic Approach

The basic idea behind our project's approach is to evaluate a descriptive/subjective answer based on a model or a sample answer provided by the teacher/examiner. The students would be required to answer questions pre-prepared on an application, and those answers will be evaluated. Each answer goes through preprocessing using natural language processing methods like stemming, lemmatization and tokenization. After the preprocessing, keywords in the model answer and the student's answer are compared using cosine similarity. The grammar of the answer is also checked. For every answer, some question related concepts are then checked using fuzzy logic. The combined results of cosine similarity and fuzzy logic are fed to the Naive Bayes algorithm which provides us with the final marks.

Algorithm

The approach towards the development of the algorithm involves amalgamating concepts of Natural Language Processing like context identification and text similarity. The ideal answer provided by the examiner and the student's answer are both taken into account. The process of Tokenization, Stemming and Lemmatization is performed. These methods are important for easier implementation of algorithms and for finding. Three parameters are specified here :

1. Keywords
2. Grammar
3. Question Related Concepts

3.3 UI Layout

UI Layout is basically the interface that will be visible to the user while using our project. This was achieved by implementing the Python Flask App which was hosted on the website application framework 'Heroku'. The main focus of a UI layout is to maximize the user's experience. This is because a good user experience means that more and more users will be able to use the application with full utilisation of the product.

An easy to use and efficient UI Layout has properties like:

Consistent

The data provided in the project should be consistent no matter how many times the user may need it. This helps in increased reliability of the project/app.

Potential development

Due to the ever-changing technologies, improvements and developments are always emerging. Hence the user may need a different layout or some extra features also. Considering this, we have modelled our project in such a way that changes can be made by the developer according to the user's choice.

Easy to use

The buttons, colours, fonts, etc. used in the project must be easy to use by the user. The fonts should be clear and the buttons should be clickable for proper user experience. The colours should also be subtle considering the eyesight of the users.

Easy layout

The layouts used in the interface should be easy to use and understandable to the user and the developer. This helps in smooth working of the application.

Input

The input that is to be provided by the user must be entered easily. This is because the user may or may not enter multiple keywords. Regardless of the number of keywords used, the classification should be achieved in a smooth manner.

3.4 Hardware Requirements

System specification:

1. Microsoft windows 10/8/7
2. RAM: 1GB(32-bit) or 2 GB (64-bit)
3. Processor : 1 GHz or faster

3.5 Software Requirements

3.5.1 Py- charm IDE Community Edition 2018.1.4 x64

3.5.2 Python 2.7 or python 3

3.5.3 NLTK package present for python

3.5.4 Scikit learn package for python

3.5.5 Heroku platform

3.6 Approach to implementation

The basic implementation approach used by us in designing of our project is given in the following points:

1. Formulation of the 'ideal' descriptive answer.

   The answer which is ideal according to the teacher is provided here. This answer is converted to a csv file for ease of comparison.
2. Descriptive Answer sheet taken as input

   The answer given by the student is taken as input in an application we made on Heroku platform

   all the sentences go through preprocessing in an orderly fashion. The preprocessing starts with stemming, followed by lemmatization and tokenization.
3. Comparison of the Two Answer Sheets

   The first parameter of comparison between the two answer sheets is on the basis of keywords. The keywords obtained by text preprocessing are taken into account, and a Cosine Similarity test is performed on them. This test gives us a key value between 0 to 6, 0 meaning that there is no similarity between the two answers and 6 meaning maximum similarity between the two answers.
4. Grammar check

   The grammatical mistakes along with the word limit of the answers are checked.
5. Question Related Concepts

Question Related Concepts and some concepts specific to each question. These concepts carry marks because they are necessary to answer a question. The question related concepts between the two answers are compared using Fuzzy Logic. This also generates a key value between 0 and 1, 0 meaning least similarity and 1 meaning maximum similarity.

6. Getting the Final Result

   The keys generated by Fuzzy Logic and Cosine Similarity are fed to Naive Bayes algorithm, which helps calculate the final marks for the student.

3.7 Implementation

The automated descriptive answer sheet evaluation process can be divided broadly into 4 stages. These are:-

- Text Summarization
- Text Preprocessing
- Similarity Measures
- Allotment of marks

3.7.1. Text Summarization

Text summarization refers to the technique of shortening long pieces of text. The intention is to create a coherent and fluent summary having only the main points outlined in the document

Algorithm of Text summarization

1. Accept the input text

2. Now tokenize the accepted text into word
3. Removal duplicates from word list
4. Put a counter for frequency of each word
5. Compute the word percentage by dividing word frequency by the length of word list
6. Compare the word percentage with a maximum and minimum threshold value and select average frequent word as keywords and remove most frequent word and less frequent word.
7. Counter for the size of window for each sentence using keywords
8. Compute weight of each sentence by dividing square of no of keyword in sentence by window size
9. Sort the sentence in a descending order based on weight value and select first n sentence as summary

3.7.2. Text Preprocessing

In the summarized texts there are certains words which carry less information and can be ignored so that further text processing tasks can be facilitated. Pre processing refers to the way in which data can be converted to computer understandable form. A very efficient way to handle text preprocessing is natural language processing. It contains tokenize text into word, removes StopWord, lemmatize word, remove duplicate word etc. Natural Language Toolkit (NLTK) is considered a leading platform for building python programs to work with human language data. It has huge amount of built in libraries which can be used in the text preprocessing by providing us with the benefit of typing less commands. The NLTK function word_tokenize, which is a built in function, is used to split the text into word and store in a list. The most crucial step in text preprocessing is to filter out the unnecessary words and remove the redundancy of the documents.NLTK contains StopWord corpus which consists of the unnecessary words which are irrelevant to define the meaning of the sentence.

Lemmatization can be defined as the way of changing words to its basic form called lemma. The purpose is to save time used in processing as the number of words in the word list are reduced. WordNet lemmatizer which is a built-in function in NLTK converts the words from word list to their basic forms. The purpose is to keep the format of all the data the same, this is required to perform some application on the data. The format can be bigram or digram i.e. sequence of the neighbouring elements in a token string. Similarity of structure of text is analyzed using the frequency distribution of bigram. Bigrams are generated using an inbuilt function in NLTK and it generates a bigram list of all words. A dictionary is used to store the word as a key and frequency count of each word. After making this dictionary of the bigrams and frequency count of each word, it is used as a way to measure similarity.

```
tokenizer = RegexpTokenizer("[\w']+")        #Function to tokenise from regular expression
lemmatizer = WordNetLemmatizer()
sentence="You would need to add materials that you need to use. You also would want to know how much vinegar you should pour in the cups. You should

arr = []
arr1 = []
sentence=tokenizer.tokenize(sentence)
```

Fig 3.2 Code snippet for tokenisation

```
english_stops = set(stopwords.words('english')) #Setting up the function for stopwords
with open('sample.csv', newline='') as f:   #To open the CSV file and read line by line
  reader = csv.reader(f)
  for row in reader:
    print(row)
    row = str(row)              #Converting line input to string so that tokeniser can process the function
    row = row.lower()               #Converting string to lower case so that stop words can be used easily.
    row = tokenizer.tokenize(row)       #Using the tokenise function
    print (row)
    print ('\n')
    words = row
```

Fig 3.3 Code snippet for Stemming

3.7.3. Similarity Measures

In many cases, it is needed to define whether two sentences are similar or not. Similarity measures is a term which tells if two sentences are similar or not by considering the different angle of similarity. Several similarity measure techniques are available that can be performed. In this experiment, cosine similarity is performed.

Cosine similarity

Cosine similarity is an efficient similarity measure technique. It looks at the angle by two documents and tells how similar they are.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}},$$

Fig 3.4 Cosine Similarity formula

Algorithm of cosine similarity

1. Input dictionary of words and frequency.
2. Two vectors are to be taken, one for the model answer and another for the student answer.

Length of Total word list should be the length of each vector.

3. Compute dot product of two vector
4. Find norm of first vector
5. Find norm of second vector
6. Multiply the first and second norm
7. To find cosine similarity divide the dot product result by multiplication result.

```
from sklearn.metrics.pairwise import cosine_similarity

dot = sum(a*b for a, b in zip(vec_a, vec_b))
norm_a = sum(a*a for a in vec_a) ** 0.5
norm_b = sum(b*b for b in vec_b) ** 0.5

cos_sim = dot / (norm_a*norm_b)
```

Fig 3.5 Code snippet for Cosine Similarity

3.7.4. Allotment of Marks

The aim of this project is to assess the descriptive answers using automation techniques and assign marks. This will lessen the time for assessing answer scripts and bring equality for evaluation. For satisfying those requirements, we have used a weighted parameter-based-technique for automatic assessment.The summary generated by the extracted text is vital to the experiment. For keeping the summary generation efficient we have generated a summary of two techniques, that is keyword based summarization and bag of word based

summarization, keeping in consideration the reference summary. The estimated score of the answers is displayed in the Table.
Table indicates that the F-score of our used summarization technique is greater than the bag-of-words based summarization technique. We have considered five parameters in this experiment for scoring marks. These are synonym similarity, bigram similarity, grammatical-spelling error, cosine similarity and Jaccard similarity.

Feature	Keyword based summarization	Bag of word based summarization
Precision	0.9	0.83
Recall	0.83	0.41
F--score	0.86	0.53

Table 3.1: Score Calculation

The above parameters are used to automatically assess two types of questions (M50and M100) based on marks. A unique value of weight is allotted to each of the parameters depending on the type of question. The value of the weight is assigned after taking an average of the model answers for each parameter. Through the model answers we have observed that the significance of grammatical and spelling error type parameters is less compared to the synonym for evaluation of answer script.

The higher the value of the weight, the more significance it holds for marks allocation. The values of the parameters can range from 0 to 1 depending on the similarity and presence of the error.Higher parameter value symbolises that similarity between the model answer and student answer is more the similarity between the two and vice versa. In this project we have taken 100 sample answers for testing the accuracy of our model. The answer script consists of two types of questions.

Further the five above mentioned parameters are computed from the 100 sample answer sheets and are used for automatic scoring. Most of the cases the automatically allocated score and manually assigned marks are very close. When the student answer and the true answer contain more structural similarity as well as synonym similarity, the automated scored marks are very close to the manually scored marks. On the other hand, a notable difference between the automated scored marks and manually scored marks exist when the student answer and the true answer have less structural similarity while more Cosine similarity.

3.8 Design and Development

The figures given below show the design process in a step by step manner.

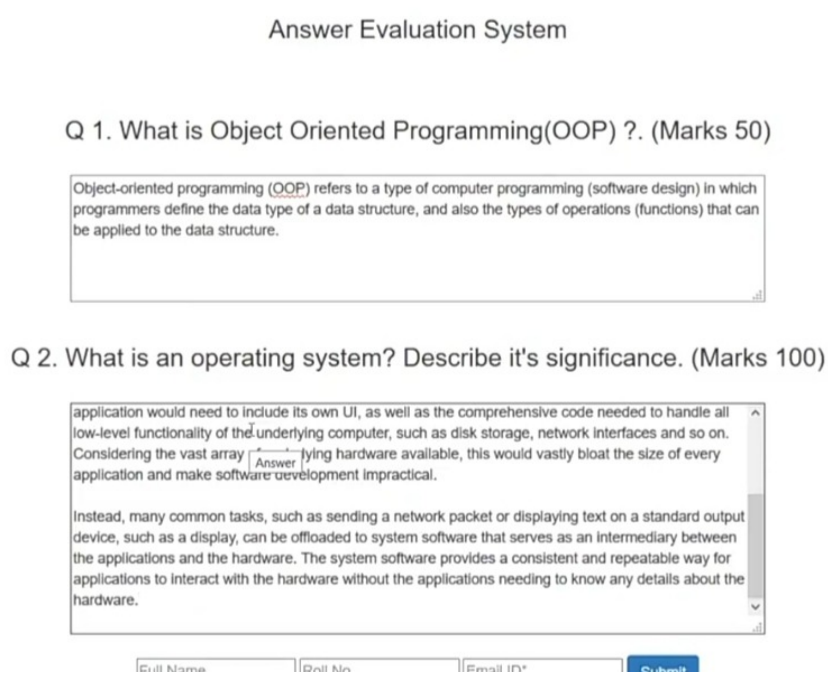

Fig3.6 Heroku Application for taking students' answer input

```
import numpy as np
import cv2
import pytesseract

img = cv2.imread(r"C:\Users\niyat\OneDrive\Desktop\sample.png")
pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files (x86)\Tesseract-OCR\tesseract.exe'
gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
gray, img_bin = cv2.threshold(gray,128,255,cv2.THRESH_BINARY | cv2.THRESH_OTSU)
gray = cv2.bitwise_not(img_bin)
kernel = np.ones((2, 1), np.uint8)
img = cv2.erode(gray, kernel, iterations=1)
img = cv2.dilate(img, kernel, iterations=1)
out_below = pytesseract.image_to_string(img)
print("OUTPUT:", out_below)
```

Fig 3.7 Code snippet for pre- processing

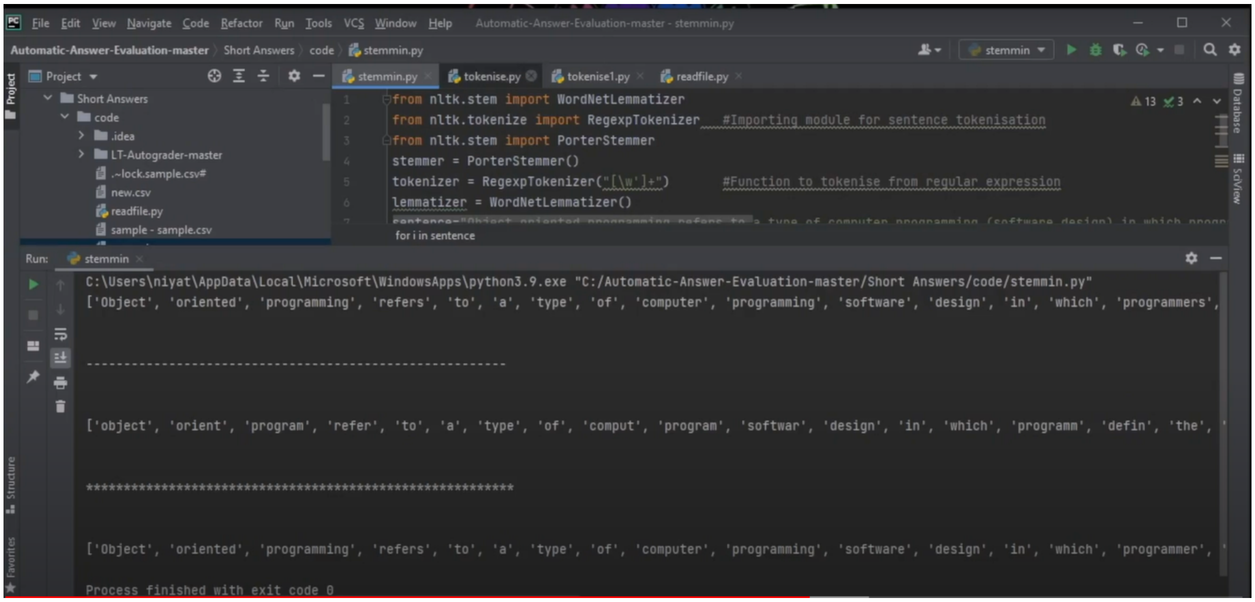

Fig 3.8 pre-processed data

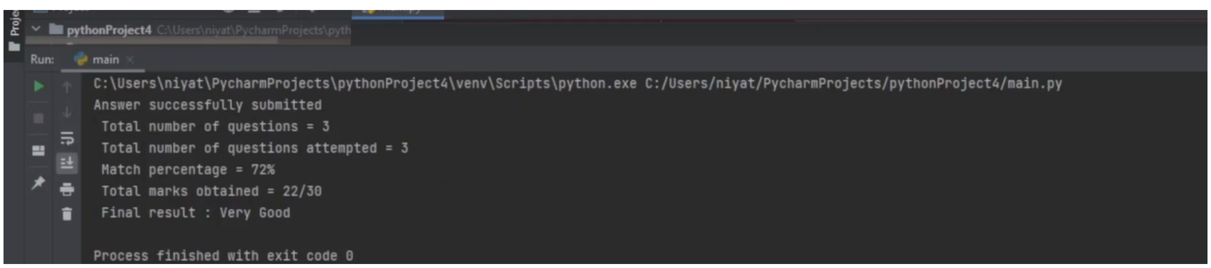

Fig 3.9 final result

4. RESULTS

The basic motive of this project was to reduce the manual work required for the evaluation of descriptive answer sheets. Examiners aren't always able to give their complete focus while checking answer sheets and sometimes may become biased while allotting marks to the students. Hence, we tried and were somewhat successful in achieving the goal of decreasing the manual load of teachers and also the unbiased allotment of marks by creating this project.

To fulfill these needs, we used weighted parameter based technique, the generation of summary played an important role and it was done using F-score. High parameter value would indicate more similarity in the two answers.

The methodology used was able to give accurate results to an extent. The answers provided by us for comparison were first checked to be correct before further evaluation of the sample set. The marks allotted to the sample set were also unbiased as they were based on the text similarities and not on the teacher's judgement.

The implementation of NLP allowed us to get precise results with precision value of 0.8 and a recall of 0.93. All possible words and their combinations were created with accuracy and helped us in the completion of the project.

5. CONCLUSION

The basic conclusion of this entire project is that we have created an easy method for users, especially teachers, to be able to evaluate answer sheets containing descriptive answers with more ease and less manual power. The user provided the answer expected and the csv file was created. Accordingly the csv file of the descriptive answer provided by the student was created and both the files were preprocessed and checked for similarity and the allotment of marks was done based on these similarities.

Here we accepted the answers in typed text format and stored them in csv files. After which summarization was performed using the keyword based technique. The summarized text was then used for pre-processing using in-built functions of NLTK; tokenization, lemmatization and stemming was performed. Grammatical and spelling errors were also included for evaluation purposes. Then this is further used to calculate parameter values using cosine similarity measure. These parameter values are used in calculating the final score by multiplying with pre specified weighted values.

We have included all the methodologies that we used in the implementation of the Automatic Descriptive Answer Evaluation project. The dataset consisted of about 150-200 answer samples. All the samples used were in the form of texts and the UI layout was also made in consideration with all the user requirements. This project helped all the group members in learning more about Machine Learning and all the new and upcoming innovations. We were also able to evaluate answers efficiently without time consuming algorithms. We found for most of the automated evaluated answers, the score was similar to the manually given score.