*Project Report On*

# Subjective Answer Evaluation Using Machine Learning

*submitted in partial fulfillment of the requirements for the award of the degree of*

## Bachelor of Technology

*in*

***Computer Science & Engineering***


**By**

**Elzaba Babu (RET18CS069)**

**Under the guidance of**
**Ms Meenu Mathew**

RSET
RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

**Department of Computer Science & Engineering**
**Rajagiri School of Engineering and Technology**
**Rajagiri Valley, Kakkanad, Kochi, 682039**


**July 2022**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**
**RAJAGIRI SCHOOL OF ENGINEERING AND TECHNOLOGY**
**RAJAGIRI VALLEY, KAKKANAD, KOCHI, 682039**

**CERTIFICATE**

*This is to certify that Project report entitled* ***"Subjective Answer Evaluation Using Machine Learning"****, report of the project presented during the eighth semester by* ***Elzaba Babu(RET18CS069)****, to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B.Tech) in Computer Science & Engineering during the academic year 2021-2022.*

**Dr. Dhanya P M**
*Head of Department*
*Dept. of CSE*
*RSET*

**Ms Jincy J Fernandez**
*Project Coordinator*
*Asst.Professor*
*Dept. of CSE*
*RSET*

**Ms Meenu Mathew**
*Project Guide*
*Asst.Professor*
*Dept. of CSE*
*RSET*

***External Examiner***

***Internal Examiner***

## ACKNOWLEDGEMENTS

I wish to express my sincere gratitude towards Dr.P.S. Sreejith, Principal, of RSET, and Dr.Dhanya P.M, Head of Department of Computer Science & Engineering for providing me the opportunity to present my project "Subjective Answer Evaluation Using Machine Learning".

I am highly indebted to my project coordinators, Ms Jincy J Fernandez, Assistant Professor, Department of CSE, Ms.Meharban M S, Assistant Professor, Department of CSE, and Mr. Harikrishnan M, Assistant Professor, Department of CSE for their valuable support.

It is indeed my pleasure and a moment of satisfaction to express my sincere thanks to my project guide Ms. Meenu Mathew, Assistant Professor, Department of CSE, for her patience and all priceless advice and for all the wisdom she has shared with me.

Last but not the least, I would like to express my sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

**Elzaba Babu**

## ABSTRACT

Every year educational institutes conduct various examinations, which include institutional and non-institutional competitive exams. Nowadays online tests and examinations are becoming popular to reduce the burden of the examination evaluation process. The online exams include either objective or multiple-choice questions. Nevertheless, the exams include only objective or multiple-choice questions. Subjective questions are the most effective way to assess a student's understanding and play an important role in determining how well a student has understood a subject. However, subjective-based questions and answers are not involved due to the evaluation process complexity and efficiency of the evaluation process. Additionally, the marks allotted to these answers may vary from examiner to examiner which leads to inconsistencies in the correction.

This raises a concern to have an automated system that eliminates bias in correction while reducing the time and effort put into it. It shall also ensure greater accuracy by minimising errors. The fundamental role of the system will be to take the answer sheets and model answers as inputs and return an unbiased and completely evaluated answer sheet as the output. It calculates the score of the student by combining various parameters such as keywords, question specific things along with the proper grammar, which in terms provides a more accurate score. The system must be capable of scoring the answer papers within the range of those awarded by human evaluators. It must be consistent in the way it grades the answer scripts and thus it can save the time and cost of evaluation.

# Contents

# List of Figures

# Abbreviations

| | | |
|---|---|---|
| **BERT** | : | Bidirectional Encoder Representations from Transformers |
| **BPE** | : | Byte-Pair Encoding |
| **ELMo** | : | Embeddings from Language Models |
| **GPT-2** | : | Generative Pre-trained Transformer 2 |
| **GUI** | : | Graphical User Interface |
| **ML** | : | Machine Learning |
| **NB** | : | Naive Bayes |
| **NLP** | : | Natural Language processing |
| **QST** | : | Question Specific Things |
| **RNN** | : | Recurrent Neural Networks |
| **RoBERTa** | : | Robustly Optimised BERT Pre-training Approach |
| **LSTN** | : | Long Short-Term memory Networks |

# Chapter 1

# Introduction

## 1.1 General Background

Subjective questions are capable of examining the adopting ability of knowledge of the student, however, correctly evaluating the answer scripts is a challenging and complex job to perform, hence an efficient mechanism is needed to tackle this challenge by not only omitting human errors but also providing a quicker and faster output.The student enters the answer in google form.Evaluator uploads both student and model answer into the system.The student and model answers are compared and based on this comparison marks are assigned.Here comparison is mainly done between the keywords of the model answer and the entire student answer .Both model and student answers undergo various preprocessing steps.After comparison the similarity between them is notified using the Jaccard similarity.Based on the similarity between the answers they are assigned to various class labels ranging from 0-10.We use a combination of keywords,grammar and qst to assign them to specific class labels.After this marks are assigned to each answers and they are combined to generate the total score.

## 1.2 Objective

The goal of this project is to develop a software to ease the burden on examiners. The answers will be typed by the students and various preprocessing steps using NLP techniques will be applied. Model answer sets will be provided by the evaluator. The user answer will be compared with the model answer sets based on the keywords and grammar. Answers will be graded and results will be displayed accordingly.

### 1.3 Motivation

It takes a lot of time for the teachers to do the corrections since they have to carefully go through each and every answer .Since teachers take more time to correct the answers the results also gets automatically delayed.Some evaluators may be partial to some students while others may not be.So this can affect the marks of the students and the students who have not written the answers properly may get a high score.Some moderators are strict enough while some are liberal.So strict moderators may assign less marks while others may provide average marks.So this also affect the marks of the students.

### 1.4 Summary of Report

The primary goal of this system is to develop a software to ease the burden on the evaluators.This automatic answer checker helps in correctly evaluating the answers and assigning accurate marks to the students based on the answers provided by them.It also helps in saving the time of evaluators and process the entire task quickly.

# Chapter 2

# Literature Survey

## 2.1 Evaluation using LSTM-RNN layer

The evaluation of answer papers considering semantics is a complex process that requires great intellectual effort from evaluators.The lack of availability of expert evaluators makes the evaluation more time consuming.In order to reduce the effort during the evaluation of answer scripts an automated system is required to grade the answer scripts correctly.The proposed sequential model consists of LSTM-RNN layer which sequentially takes the glove vector representation in a sentence of each word and converts to embedding vector representation.Embedding vector corresponding to the glove vector of the last word will be the representation of the entire sentence in its semantic form. The sequential model consists of embedding layer, Long Short Term Memory layer, dropout layer and dense layer. The regularisation technique, dropout reduces overfitting by preventing complex co-adaptations on training data.

## 2.2 Evaluation using Text mining

This method of Evaluation of student answers uses natural language processing and artificial neural networks. The motivation behind Text Mining is to process unstructured (textual) data, remove significant numeric files from the content, and, along these lines, make the data contained in the content open to the different information mining (statistical and machine learning calculations. Data can be extricated to determine outlines for the words contained in the archives or to word documents for the records dependent on the words contained in them. Content mining is utilised to separate vital data or information or example or learning from the test proprietors and applicants' answers which are in the unstructured frame. The basic reason behind text mining is to find helpful information from natural language text. After the text mining is applied the words like am, is,are, was etc. are eliminated by NLP algorithms and we get keywords from the answer. After identification of the keywords the system shows the total number of keywords to the exam owner. According to the quantity of catchphrases, the test administrator can choose a checking plan for that specific answer.

### 2.3 Evaluation using BERT

Word embeddings that consider context have attracted great attention for various natural language processing tasks in recent years. This paper utilises contextualised word embeddings with the transformer encoder for sentence similarity modelling in the answer selection task. Two different approaches (feature-based and fine-tuning-based) for answer selection are being used. In the feature-based approach, it utilises two types of contextualised embeddings, namely the Embeddings from Language Models (ELMo) and theBidirectional Encoder Representations from Transformers (BERT) and integrate each of them with the transformer encoder. It found that integrating these contextual embeddings with the transformer encoder is effective to improve the performance of sentence similarity modelling. In the second approach,it fine-tunes two pre-trained transformer encoder models for the answer selection task. Based on the experiments on six datasets, it finds that the fine-tuning approach outperforms the feature-based approach on all of them. Among the fine-tuning-based models, the Robustly Optimised BERT Pre Training Approach (RoBERTa) model results in new state-of-the-art performance across five datasets.

### 2.4 Evaluation using RoBERTa Fine tuning

The BERT model was significantly undertrained. The RoBERTa model was proposed by modifying different hyperparameters in BERT along with new design choices. More specifically, RoBERTa used much larger mini batches and learning rates compared to BERT. Also, the next sentence prediction task was removed from the pre-training stage. Five different datasets were used for pre-training, which in total consists of around 160GB of un-compressed text. These new parameter settings and objectives showed significant improvements in the BERT model in different NLP tasks. To fine-tune it for the answer selection task,it followed the similar approach of BERT fine-tuning by modifying the final layer to utilise it for similarity modelling.

### 2.5 Naive Bayesian Classification

The naive Bayes classifier greatly simplifies learning by assuming that features are independent given class. Although independence is generally a poor assumption, in practice naive Bayes often competes well with more sophisticated classifiers. It analyses the impact of the distribution entropy on the classification error, showing that low-entropy feature distributions yield good performance of naive Bayes.

Bayesian classifiers assign the most likely class to a given example described by its feature vector. Learning such classifiers can be greatly simplified by assuming that features are independent given class, that is, $P(X|C)=\prod_{i=1}^{n} P(X_i|C)$, where $X = (X_1, X_2, ........., X_n)$ is a feature vector and C is a class. Despite this unrealistic assumption, the resulting classifier known as Naive Bayes is remarkably successful in practice, often competing with much more sophisticated techniques. Naive Bayes has proven effective in many practical applications, including text classification, medical diagnosis,and systems performance management.

# Chapter 3

# Proposed Method

## 3.1 Problem Definition

Subjective questions are capable of examining the adopting ability of knowledge of the student, however, correctly evaluating the answer scripts is a challenging and complex job to perform, hence an efficient mechanism is needed to tackle this challenge by not only omitting human errors but also providing a quicker and faster output.

## 3.2 Scope of the work

- Automate the evaluation process for subjective answers using ML
- A system for descriptive answer checking and grading.
- Allocates marks to the user after verifying the answers for online tests and examinations
- Answers will be checked with the model answer given by the paper setter.

## 3.3 Methodology

### 3.3.1 Selection of Data sets

We had developed a dataset which consists of 50 students and 20 questions.Each student enters answers for these 50 questions.Then we generate a list which consists of the answers of 50 students for the first question, similarly the second question and so on.

#### 3.3.2 Selection of prepossessing steps

Preprocessing involves converting the text to lower case, removing special characters using regular expression, removing punctuations and removing white spaces

#### 3.3.3 Selection of model

We have chosen RoBERTa for tokenizing the sentences. RoBERTa stands for Robustly Optimised BERT Pre-training Approach. It was presented by researchers at Facebook and Washington University. The goal of this method is to optimise the training of BERT architecture in order to take less time during pre-training.

### 3.4 System Architecture

This system consists of a user interface which is developed using Tkinter. Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

The user interface is used by the evaluator. It provides the evaluator with options to input the model answer and student answer and evaluate it. Evaluator will upload the student answer and model answer as two separate csv files.Model answer consists of answer of each question(In total 20 questions are there.) and corresponding keywords.Student answer consists of answers provided by each student( In total 50 students are there.) to all the questions.After evaluating it using the application, the result will be saved as a csv file which is saved directly to the directory.

- Students write answers in Google form.
- The evaluator uploads both the model and the student answer as a csv file..
- Both the model and the student answers are processed to provide the marks.
- Finally the marks of each student is viewed by the evaluator.

Evaluation is done by taking into consideration three different aspects of the student answer.They are keyword matching,QST evaluation and Grammar checking.

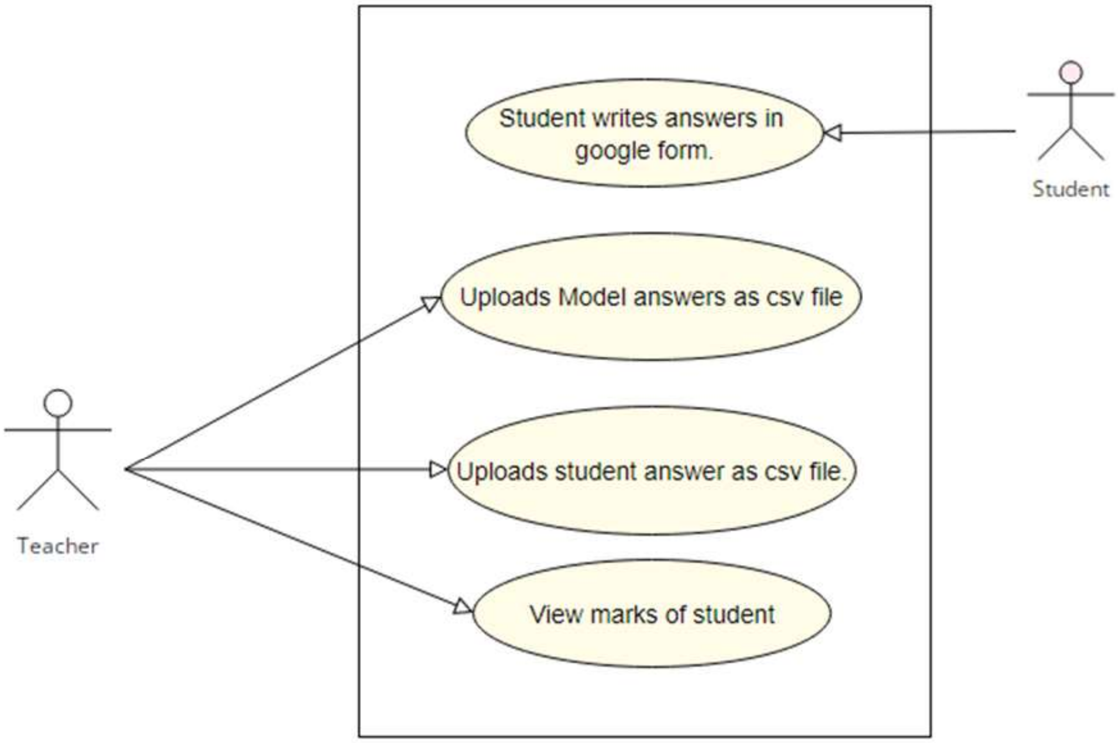



Figure 3.1: Use Case Diagram

Keywords are a great way to show students what the evaluator is intending for them to write about in an answer.The keywords for each question will be provided by the evaluator itself.If those keywords are present in the student answer then marks will be awarded based on the keyword ratio.

Keywords are extracted from the student answer using RoBERTa Tokeniser.RoBERTa model works in the same way as BERT model.It builds on BERT and modifies key hyperparameters, removing the next-sentence pretraining objective and training with much larger mini-batches and learning rates.RoBERTa has the same architecture as BERT, but uses a byte-level BPE as a tokenizer (same as GPT-2) and uses a different pre training scheme.RoBERTa doesn't have token_type_ids, you don't need to indicate which token belongs to which segment. Just separate your segments with the separation token tokenizer.sep_token.The student answer will be tokenized and corresponding token ids will be generated. In the same way token ids will be generated for the tokenized keywords from model answers as well.Both will be checked against each other to award marks for keyword matching.

QST evaluation refers to checking the similarity between 2 documents.In this work, we chose Jaccard similarity coefficient to compute the similarity between two sentences. Jaccard similarity coefficient is defined as the size of the intersection set divided by the size of the union set, ie,

$$J(A, B) = |A \cap B|/|A \cup B| \tag{3.1}$$

Grammar checking is done using Language tool python.LanguageTool is an open-source grammar tool, also known as the spell checker for OpenOffice. This library allows you to make to detect grammar errors and spelling mistakes through a Python script or through a command-line interface.It provides ruleId, a message to the end user regarding the error, the suggested replacements, the context which is the input, the offset which is the position of the start of the issue, the errorLength which is the number of characters of the issue, the category of the mistake and the releIssueType.Finally the number of error determine the score for grammar.The combination of these three factors makes up the score.

This score is passed on to the naive bayes classifier.Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.It involves a preprocessing step in which we prepare the data to be fed into the classifier.we load the dataset into our program using "dataset = pd.read_csv('user_data.csv'). The loaded dataset is divided into training and test set, and then we have scaled the feature variable.After the preprocessing step, we will fit the Naive Bayes model to the Training set.In this paper we have used the GaussianNB classifier to fit it to the training dataset. Then it will predict the test set result. For this, it will create a new predictor variable y_pred, and will use the predict function to make the predictions.

The final score is the score that comes after prediction through naive bayes classifier.All the marks for all the students for all the questions will be saved to a csv file which will directly move to a directory.

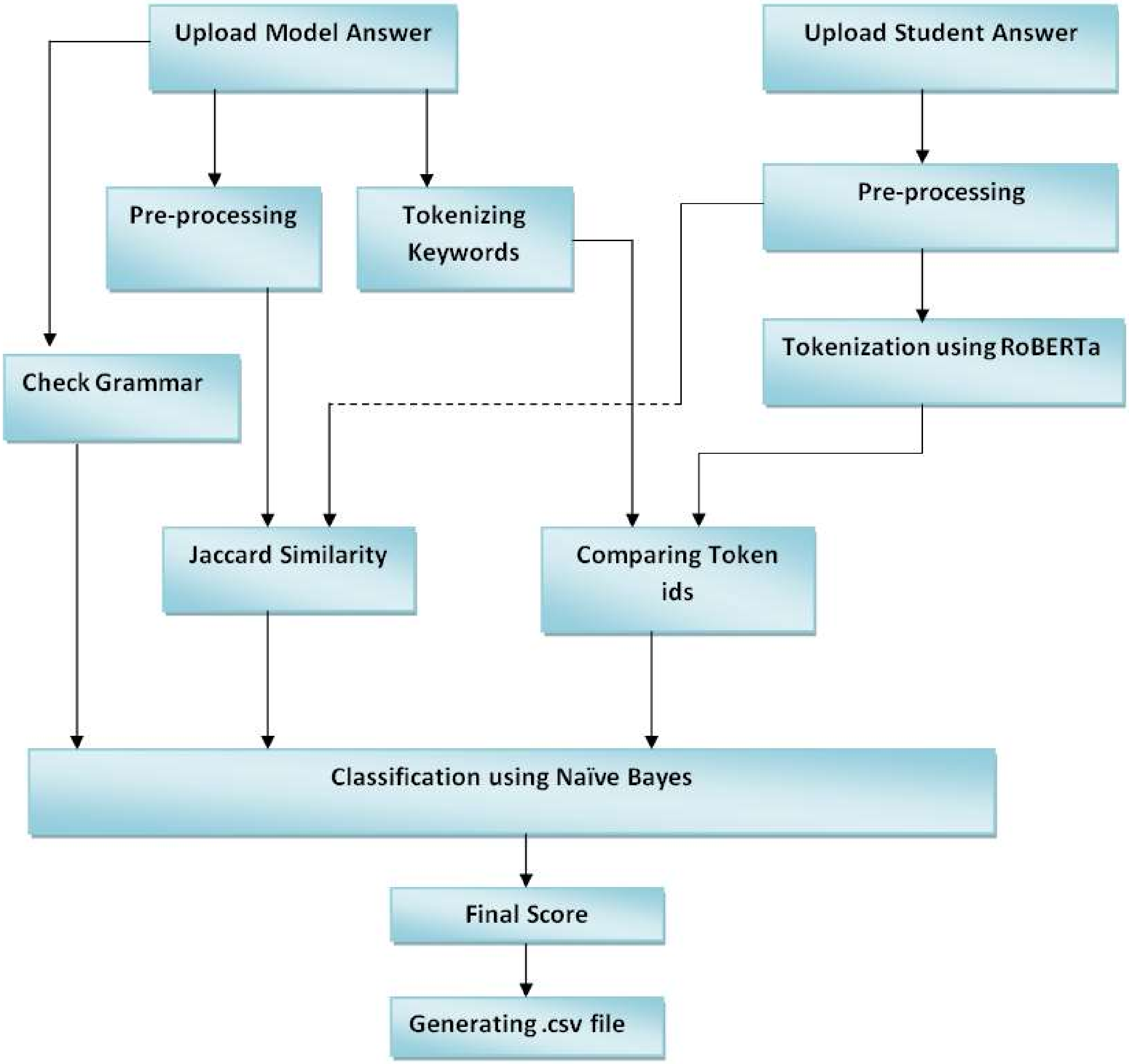



Figure 3.2: System Architecture

## 3.5 Module Division

### 3.5.1 Preprocessing

The preprocessing steps involve converting the text to lowercase character.After this conversion the special characters are removed using regular expression.Then punctuations and white spaces are removed to make it in a clean format.

### 3.5.2 Tokenization using RoBERTa

This module tokenization is done using the Roberta Model which is the robustly optimised pertaining approach to generate specific tokens.Then respective tokens Ids are also generated.It is based on Google's Bert model.It has the same architecture as that of the Bert model.

### 3.5.3 Grammar Checking

Grammar checking is done using language tool python.Tenses,spelling errors etc are detected in this phase.It is an open source grammer tool.It specifies various details of the errors and the length of the error.It allows to detect grammar errors and spelling mistakes through python scripts.

### 3.5.4 QST Evaluation

QST(Question Specific Things ) is evaluated using the jaccard similarity.Jaccard similarity is used to check the similarity between documents.It does not count the repeated words in a sentence which proves to be one of its advantage.Text distance,a python library compares the distance between 2 documents.

### 3.5.5 Classification

In the classification phase Naive Bayes classifier is used to classify into specific labels.The labels range from 1-9.These labels are used to predict the score.Highest label indicates the student has got the highest marks.Assigning into specific labels is done by evaluating the combination of keywords,grammar and qst.This phase is based on Bayes theorem.

#### 3.5.6 GUI

Final phase is the GUI. Tkinter is used to create the GUI.It is the standard library for python.Text alignments,font alignments etc can be easily done by Tkinter.

# Chapter 4

# Results and Discussions

We have created a GUI using Tkinter for teachers to upload the model answer file and student answer file. Inputs are given in the form of csv files. Output of the model is a csv file containing the marks obtained by all students for each question and their total marks.

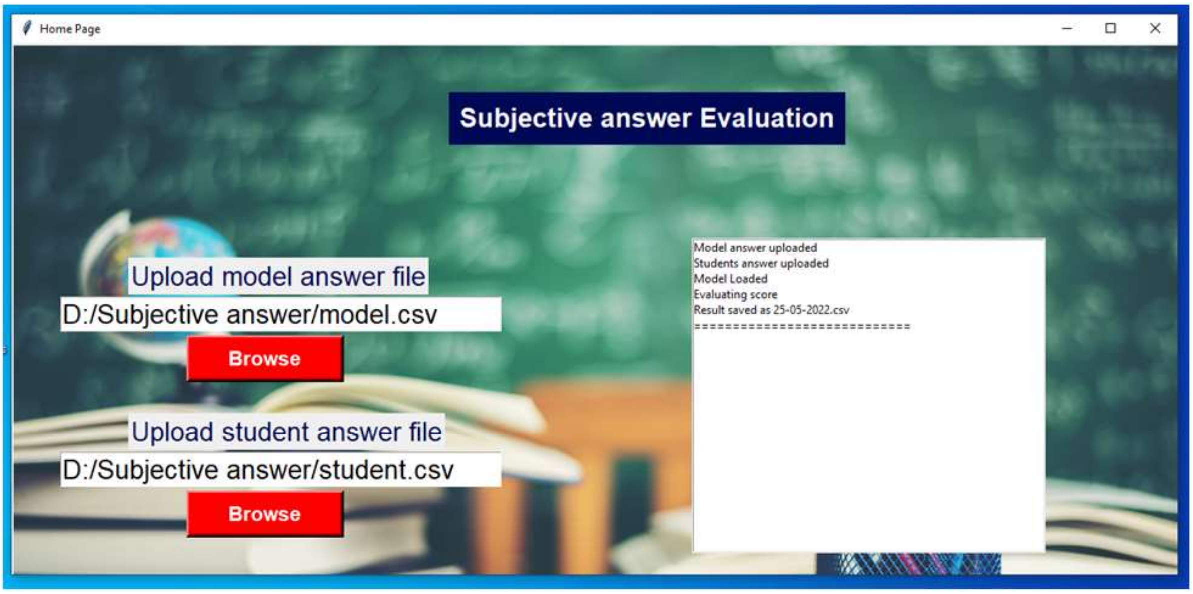



Figure 4.1: Graphical User Interface

Total number of students was 100. We have given 20 questions to each student. Each question carries 10 marks. Each student enters answers for these 20 questions. Our model evaluates each answer and generates a list which consists of the answers of 50 students for the first question, similarly the second question and so on. The model also calculates the total marks obtained by each student by adding the marks assigned for each answer.

```
======================
[1859, 12, 5400]
[26534, 33832]
[8098, 27648]
[6680, 9, 44999]
[12821, 9421]
[37328]
[1345, 18986, 1683]
[17997, 20234]
Model answer ::  Albert Einstein was a German-born theoretical physicist, who developed the special and general theories of relativity and won the Nobel Prize for Physics in 1921 fo
User answer ::  Einstein was an English theortical physicist whose theory of exploding black holes drew upon both relativity theory and quantum mechanics. He also worked with space-
[364, 40335, 21, 41, 47510, 5, 2723, 3569, 33832, 1060, 6680, 9, 28976, 909, 6538, 4855, 2115, 258, 44999, 6680, 8, 17997, 20234, 37, 67, 1006, 19, 980, 86, 23429, 2192]
user_clean_ans text==> einstein was an english theortical physicist whose theory of exploding black holes drew upon both relativity theory and quantum mechanics he also worked with
3
0
1
ypred [6]
======================
```

Figure 4.2: Evaluation of a single answer by the model.

| | Student | Question 1 | Question 2 | Question 3 | Question 4 | Question 5 | \ |
|---|---|---|---|---|---|---|---|
| 0 | Angelina | 10 | 3 | 9 | 10 | 7 | |
| 1 | Ann | 5 | 6 | 3 | 2 | 9 | |
| 2 | Annabelle | 5 | 5 | 10 | 5 | 8 | |
| 3 | Annie | 5 | 6 | 10 | 10 | 5 | |
| 4 | Antonio | 3 | 10 | 10 | 10 | 5 | |
| 5 | Arden | 10 | 10 | 10 | 10 | 10 | |
| 6 | Ariana | 10 | 8 | 2 | 5 | 5 | |
| 7 | Asees | 6 | 5 | 3 | 5 | 6 | |
| 8 | Ashly | 10 | 8 | 5 | 10 | 6 | |
| 9 | Austin | 10 | 5 | 7 | 9 | 3 | |
| 10 | Bella | 5 | 6 | 10 | 10 | 5 | |
| 11 | Betty | 10 | 5 | 10 | 10 | 8 | |
| 12 | Billie | 10 | 5 | 5 | 5 | 3 | |
| 13 | Bobbie | 3 | 6 | 10 | 10 | 6 | |
| 14 | Bony | 10 | 8 | 10 | 2 | 8 | |
| 15 | Caityln | 5 | 8 | 9 | 5 | 6 | |
| 16 | Camila | 5 | 3 | 9 | 10 | 5 | |
| 17 | Caroline | 10 | 5 | 3 | 10 | 10 | |
| 18 | Catherine | 6 | 5 | 2 | 5 | 7 | |
| 19 | Charley | 5 | 5 | 10 | 2 | 8 | |
| 20 | Cynthia | 10 | 8 | 10 | 10 | 10 | |
| 21 | Denise | 5 | 10 | 3 | 10 | 8 | |
| 22 | Devin | 5 | 6 | 3 | 10 | 7 | |
| 23 | Eden | 3 | 5 | 10 | 10 | 10 | |
| 24 | Elaine | 10 | 10 | 9 | 10 | 6 | |
| 25 | Elliot | 10 | 8 | 10 | 10 | 6 | |
| 26 | Evan | 5 | 3 | 7 | 5 | 10 | |
| 27 | Francis | 3 | 6 | 5 | 10 | 10 | |
| 28 | Hailey | 10 | 5 | 3 | 5 | 10 | |

Figure 4.3: Output 1

| | Question 18 | Question 19 | Question 20 | total |
|---|---|---|---|---|
| 0 | 10 | 8 | 10 | 171 |
| 1 | 10 | 8 | 5 | 145 |
| 2 | 3 | 6 | 10 | 142 |
| 3 | 10 | 10 | 7 | 162 |
| 4 | 10 | 10 | 5 | 162 |
| 5 | 10 | 10 | 5 | 195 |
| 6 | 3 | 5 | 0 | 131 |
| 7 | 3 | 8 | 3 | 121 |
| 8 | 3 | 8 | 2 | 136 |
| 9 | 3 | 5 | 3 | 137 |
| 10 | 10 | 10 | 0 | 160 |
| 11 | 10 | 10 | 2 | 164 |
| 12 | 10 | 5 | 10 | 141 |
| 13 | 10 | 8 | 10 | 158 |
| 14 | 3 | 10 | 5 | 147 |
| 15 | 10 | 8 | 2 | 149 |
| 16 | 10 | 6 | 5 | 152 |
| 17 | 10 | 8 | 0 | 153 |
| 18 | 3 | 5 | 5 | 139 |
| 19 | 10 | 8 | 3 | 148 |
| 20 | 10 | 10 | 10 | 171 |
| 21 | 3 | 8 | 0 | 144 |
| 22 | 10 | 10 | 2 | 149 |
| 23 | 10 | 5 | 3 | 151 |
| 24 | 3 | 6 | 2 | 147 |
| 25 | 3 | 8 | 3 | 163 |
| 26 | 3 | 8 | 2 | 139 |
| 27 | 3 | 10 | 3 | 154 |

Figure 4.4: Output 2

# Chapter 5

# Conclusion and Future Scope

## 5.1 Conclusion

We proposed a Subjective Answer Evaluation System for answer checking and grading based on Machine learning. The model assigns marks to subjective questions based on keyword matching, Grammar check, and Jaccard similarity against model answers provided by the faculty and student answer. The Naïve Bayes model is trained using combinations of three values which are keyword matching score, Grammar score and QST evaluation score and their corresponding class labels.

The project works with the same factors which an actual human being considers while evaluating grammar, presence of keywords, and presence of QST. It is concluded that using ML techniques will give satisfactory results due to holistic evaluation. This system offers a reliable, robust, and obvious short response time result. The accuracy of the evaluation can be increased by feeding it a huge and accurate training dataset.

## 5.2 Recommendation

- Further improvement by taking feedback from all the stakeholders such as students and teachers can improve the system meticulously.
- To improve the performance of the model, a significantly large and accurate training dataset can be used to train the model.

### 5.3 Scope of future work

The research for evaluating subjective answers using computers has been ongoing for more than a decade. The project has the future scope of providing an extension software to google forms to evaluate short descriptive answers. The ability to apply machine learning to evaluate descriptive answers is a promising one. However, there exists enormous ambiguity while processing natural language.

# References

[1] Jagadamba G and Chaya Shree G, *"Online Subjective answer verifying system Using Artificial Intelligence,"* Proceedings of the Fourth International Conference on I-SMAC DVD Part Number:CFP20OSV-DVD, IEEE, Oct 2020.

[2] Piyush Patil, ,Sachin Patil, Vaibhav Miniyar and Amol Bandal, *"Subjective Answer Evaluation Using Machine Learning"*, International Journal of Pure and Applied Mathematics, Volume 118, No. 24, 2018.

[3] Shreya Singh, Omkar Manchekar, Ambar Patwardhan, Prof. Uday Rote, Prof. Sheetal Jagtap and Dr. Hariram Chavan, *"Tool for Evaluating Subjective Answers using AI (TESA),"* IEEE International Conference on Communication information and Computing Technology (ICCICT), pp. 1-6, 2021.

[4] M. T. R. Laskar, X. Huang, and E. Hoque, "Contextualized embeddings based transformer encoder for sentence similarity modeling in answer selection task," Proceedings of the 12th Language Resources and Evaluation Conference, pages 5505–5514, 2020.

[5] Abhishek Girkar, Mohit khambayat, Ajay Waghmare and Supriya Chaudhary, "Subjective Answer Evaluation using Natural Language Processing and Machine Learning," International Research Journal of Engineering and Technology (IRJET), Vol-8, Issue-4, Apr 2021.

# Glossary

**RoBERTa:** A robustly optimized method for pretraining natural language processing (NLP) systems that improves on Bidirectional Encoder Representations from Transformers, or BERT, the self-supervised method released by Google in 2018.

**Naive Bayes Classifier:** Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

**Jaccard Similarity:** The Jaccard similarity index (sometimes called the Jaccard similarity coefficient) compares members for two sets to see which members are shared and which are distinct. It's a measure of similarity for the two sets of data, with a range from zero percentage to hundred percentage. The higher the percentage, the more similar the two populations

**Tokenization:** Tokenization is the process of replacing sensitive data with unique identification symbols that retain all the essential information about the data without compromising its security.

**BERT:** Bert-Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based machine learning technique for natural language processing (NLP) pre-training developed by Google. BERT was created and published in 2018 by Jacob Devlin and his colleagues from Google. In 2019, Google announced that it had begun leveraging BERT in its search engine, and by late 2020 it was using BERT in almost every English-language query. A 2020 literature survey concluded that "in a little over a year, BERT has become a ubiquitous baseline in NLP experiments", counting over 150 research publications analyzing and improving the model.

**ELMo:**ELMo is an NLP framework developed by AllenNLP. ELMo word vectors are calculated using a two-layer bidirectional language model (biLM). Each layer comprises forward and backward pass. Unlike Glove and Word2Vec, ELMo represents embeddings for a word using the complete sentence containing that word. Therefore, ELMo embeddings are able to capture the context of the word used in the sentence and can generate different embeddings for the same word used in a different context in different sentences.

**GPT-2:**Generative Pre-trained Transformer 2 (GPT-2) is an open-source artificial intelligence created by OpenAI in February 2019.GPT-2 translates text, answers questions, summarizes passages,and generates text output on a level that, while sometimes indistinguishable from that of humans,can become repetitive or nonsensical when generating long passages. It is a general-purpose learner; it was not specifically trained to do any of these tasks, and its ability to perform them is an extension of its general ability to accurately synthesize the next item in an arbitrary sequence.GPT-2 was created as a "direct scale-up" of OpenAI's 2018 GPT model, with a ten-fold increase in both its parameter count and the size of its training dataset.

**GUI:**A graphics-based operating system interface that uses icons, menus and a mouse (to click on the icon or pull down the menus) to manage interaction with the system. Developed by Xerox, the GUI was popularized by the Apple Macintosh in the 1980s. At the time, Microsoft's operating system, MS-DOS, required the user to type specific commands, but the company's GUI, Microsoft Windows, is now the dominant user interface for personal computers (PCs). A comprehensive GUI environment includes four components: a graphics library, a user interface toolkit, a user interface style guide and consistent applications. The graphics library provides a high-level graphics programming interface. The user interface toolkit, built on top of the graphics library, provides application programs with mechanisms for creating and managing the dialogue elements of the windows, icons, menus, pointers and scroll bars (WIMPS) interface.

**ML:**Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.Recommendation engines are a common use case for machine learning. Other popular uses include fraud detection, spam filtering, malware threat detection, business process automation (BPA) and Predictive maintenance.

**RNN:**A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed or undirected graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs.This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. Recurrent neural networks are theoretically Turing complete and can run arbitrary programs to process arbitrary sequences of inputs

**LSTM:**LSTM stands for Long-Short Term Memory. LSTM is a type of recurrent neural network but is better than traditional recurrent neural networks in terms of memory. Having a good hold over memorizing certain patterns LSTMs perform fairly better. As with every other NN, LSTM can have multiple hidden layers and as it passes through every layer, the relevant information is kept and all the irrelevant information gets discarded in every single cell.

# Appendix A

# Base Paper

**International Journal of Pure and Applied Mathematics**
**Volume 118 No. 24 2018**
ISSN: 1314-3395 (on-line version)
**url:** http://www.acadpubl.eu/hub/
Special Issue
http://www.acadpubl.eu/hub/

# Subjective Answer Evaluation Using Machine Learning

Piyush Patil [1],Sachin Patil [2] ,
Vaibhav Miniyar[3],Amol Bandal[4]
[1,2,3,4]Department of Computer Engineering,
Sinhgad Institute of Technology, Lonavala,India
piyushpatil666@gmail.com[1],
sachin.patil521@gmail.com [2],
vaibhavminiyar@gmail.com[3],
amolbd1987@gmail.com[4]

May 23, 2018

**Abstract**

The current way of checking subjective paper is adverse. Evaluating the Subjective Answers is a critical task to perform. When human being evaluates anything, the quality of evaluation may vary along with the emotions of Person. In Machine Learning, all result is only based on the input data provided by the user. Our proposed system uses machine learning and NLP to solve this problem. Our Algorithm performs a task like Tokenizing words and sentences, Part of Speech tagging, Chunking, Chinking, Lemmatizing words and Wordnetting to evaluate the subjective answer. Along with it, our proposed algorithm provides the semantic meaning of the context. Our System is divided into two modules. The first one is extracting the data from the scanned images and organizing it in the proper manner and the second is applying ML and NLP to the text retrieved from the above step and giving marks to them.

***Key Words*:**Nave bayes, Cosine Similarity, Classifier, Semantic Checking, Machine Learning

# 1 Introduction

The manual system for evaluation of Subjective Answers for technical subjects involves a lot of time and effort of the evaluator. Subjective answers have various parameters upon which they can be evaluated such as the question specific content and writing style. Evaluating subjective answers is a critical task to Perform. When human being evaluates anything, the quality of evaluation may vary along with the emotions of the person. Performing evaluation through computers using intelligent techniques ensures uniformity in marking as the same inference mechanism is used for all the students. In Machine Learning, all result is only based on the input data provided by the user. Our Proposed System uses machine learning and NLP to solve this problem. Our Algorithm performs a task like Tokenizing words and sentences, Part of Speech tagging, Chunking, chinking, Lemmatizing words and Wordnetting to evaluate the subjective answer. Along with it, our proposed algorithm provides the semantic meaning of the context. Our System is divided into two modules, Extracting the data from the scanned images and organizing it in the proper manner and Applying ML and NLP to the text retrieved from the above step and giving marks to them. The software will take a scanned copy of the answer as an input and then after the preprocessing step, it will extract the test of the answer. This text will again go through processing to build a model of keywords and feature sets. Model answer sets and keywords categorized as mentioned will be the input as well. The classifier will then, based on the training will give marks to the answers. Marks to the answer will be the final output. The need for online examination aroused mainly to overcome the drawbacks of the existing system. The main aim of the project is to ensure user-friendly and more interactive software to the user. The online evaluation is a much faster and clear method to define all the relevant marking schemes. It brings much transparency to the present method of answer checking The answers to all the questions after the extraction would be stored in a database. The database is designed as such that it is very easily accessible. Automating repetitive tasks has been the main aim of the industrial and technological revolution. The work of checking hundreds of answer sheets which more or less contains the same answer can be quite a boring task for the teachers. This system can be used instead in order to reduce their burden. It will save a lot of effort and time on teachers part. The human efforts applied in this repetitive task can be saved and spent more in other academic endeavors. The obvious human mistakes can be reduced to obtain an unbiased result. The system calculates the score and provides results fairly quickly. This system can be widely used in academic institutions such as schools, colleges, coaching and institutes for checking answer sheets. It can also be implemented in different organizations which conduct competitive examinations.

The software will take scanned copy of the answer as an input and then after the preprocessing step it will extract the test of the answer. This text will again go through processing to build a model of keywords and feature sets. Model answer sets and keywords categorized as mentioned will be the input as well. Classifier will then, based on the training will give marks to the answers. Marks to the answer will be the final output.

The paper is organized as follows: Section II contains the review of related work. Section III gives brief idea about working of system. Section IV contains Experimental Analysis and section V contain the conclusions of this research work.

## 2 LITERATURE SURVEY

Evaluation of subject answer checking isnt a new thought. It has been in the works since a decade and a half. A large number of techniques where experimented with to solve the problem efficiently. Natural Language processing, Latent Semantic Analysis, Generalized Latent Semantic Analysis, Bayes theorem, K- nearest neighbor, etc. In general they can be categorized as follows : Clustering techniques, classification techniques and natural language processing techniques.Intelligent Essay evaluator developed by Landauer[3],[4-7] in 2003 using a technique known as Latent Semantic Analysis. It gives results in the accuracy range of 60-90 %. A slightly better version of using the probabilistic LSA technique[8-10] used to develop automatic essay evaluator tool by Kakkonen. Generalized LSA[11] technique extends the LSA approach by working on vectors(n-gram, bag of vectors) instead of the dual document-term representation.

It gave a better accuracy upto 96% .

Along with the above clustering methods, classification methods such as Bayes theorem[12], K-nearest neighbour[13], Maximum entropy[14], etc were also experimented with. Bayes theorem used by Rudner in 2002 has an accuracy of 80%. The clustering technique of K nearest neighbour is based on random selection of cluster heads and then carving out clusters based on their distances from those heads. It produced results with 76% accuracy. A tool named as C-rater which uses the Maximum Entropy technique for evaluation of short answers. It gives an 80% accuracy with the score assigned by a human-grader. One major natural language processing technique we ought to look is BLEU (bilingual evaluation understudy)[19-20] is a basically an algorithm for evaluation of the text quality which has been translated with the help of a machine from one language to another. Though it is built to mimick human evaluations at a corpus level, it has a bad performance if it is used to evaluate the quality of individual sentences, which explains the poor accuracy of 50%.

# 3 WORKING

This system can be widely used in academic institutions such as schools, colleges, coaching and institutes for checking answer sheets. It can also be implemented in different organizations which conduct competitive examinations. Our Algorithm performs a task like Tokenizing words and sentences, Part of Speech tagging, Chunking, chinking, Lemmatizing words and Wordnetting to evaluate the subjective answer. Along with it, our proposed algorithm provides the semantic meaning of the context.

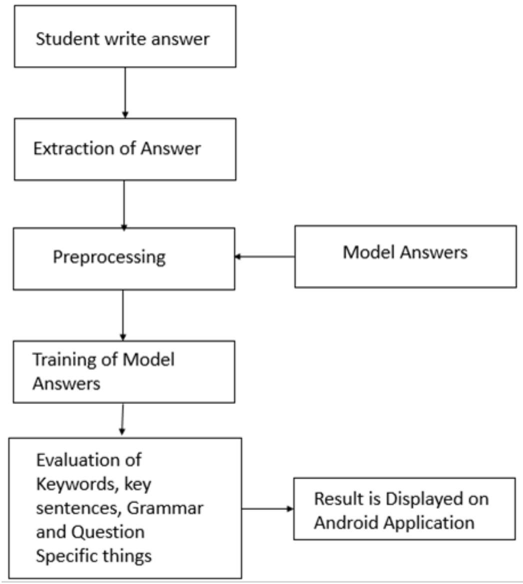



Fig. 1-Workflow diagram

Student writes answer on answer-sheet.The system will take scanned copy of the answer as an input and then afterthe preprocessing step it will extract the text of the answer. Model answer sets will be provided by the moderator/evaluator. This model answers will be then trained. The evaluator also provides with the keywords and question specific things(QST). Model answer sets and keywords categorized as mentioned will be the input aswell. Now the answer text extracted from the student ie. the user answer will be searched for the presence of keywords, QST, grammar and will be categorized and named internally as provided in table1. Grammar will be checked with an api and internalized as given in table 2. Nave bayes is used as a based classifier in our system.Nave bayes is based on three parameters i.e. Keywords, Grammar and Question Specific things.

*Mathematical model:*

P(Class — Keyword,Grammar,QST) = P(Keywords — Class) *P(grammar — class)*P(QST — class)*P(class)

For example If we have the values of Keywords,grammar,QST as 2,0,2 respectively. Then we can evaluate the class. For the input values given, it is evaluated against all the classes and then the class having the maximum probability is the class to which the given input will belong.

Each Answer will be given biased value in between 1-10. Actual marks will be evaluated on that basis. For example, If after evaluation 6 is the value we are getting and question is of 5 Marks then the actual marks calculated from this using -

Marks obtained for the answer out of 10 = 6 * 5 /10 = 3 i.e. generically we can define the formula as

Actual Marks=(Biased Value After Evaluation) * (Max marks that can be obtained for the answer/ 10)

As the dataset in ML approaches works well with numeric dataset we have mapped our six values as follows:-

| QST and Keyword Values | Non-numeric values | Numeric values |
|---|---|---|
| Excellent | E | 1 |
| Very Good | Vg | 2 |
| Good | G | 3 |
| Ok | O | 4 |
| Poor | P | 5 |
| Very poor | Vp | 6 |

Table. 1-Handling Non-Numeric Values of Keywords and key sentences

| Grammar Values | Numeric Values |
|---|---|
| Proper | 1 |
| Improper | 0 |

Table. 2-Handling Non-Numeric Values of Grammer

These 3 values i.e. Keywords, Grammar and Question Specific things is passed to Nave bayes classifier as a input. Naive bayes classifier is probabilistic classifier which is based on the maximum probability to which the given input belongs.

Now inorder to evaluate these 3 parameters we are using following strategy :-

i]Keywords and key sentences: (e, vg, g, o, p, vp)

In Cosine Similarity first we will make the vector of both model answer and users answer.We transfer the answer into vector form using cosine method.Lesser the Angle greater the similarity and greater the value of cos$\theta$.

It is calculated using two components numearator(num) and denominator(den)

Num = $\sum$(vec1[x] * vec2[x])

Where vec1[x] andvec2[x] are answer vectors and model answers vector respectively.

Den= $\frac{sum1}{den} * \sqrt{sum2}$

Where sum1,sum2 are keys obtained from model answer and user answer.

ii]Grammar: (y, e) - API which gives number of errors in the answer. This is evaluated only if above phase has some value. For Improper Grammar: 0, For Correct Grammar: 1

iii]Question Specific things: (e, vg, g, o, p, vp).Here we are using Fuzzy wuzzy - using multiple ratio functions available in Fuzzy Logic.The Fuzzywuzzy library analyses the text using degrees/features of text instead of the rigid Boolean values of 0 or 1.

After having some observations we have 21 of them as our training dataset.We will get any one of 1-9 value as the output from this Classifier. NB classifier gives the output with evidence/ probability value. Further Depending on that value we can increase the accuracy

Example: if output is 6 and the probability that given query belongs to 6 is 70% then we can increase this 6as $6 + 0.7 = 6.7$. So the total marks evaluated on the basis of 6.7.

| Keywords | Grammer | QST | Class |
|---|---|---|---|
| 1 | 1 | 1 | 9 |
| 1 | 0 | 1 | 9 |
| 2 | 0 | 1 | 8 |
| 1 | 1 | 2 | 8 |
| 1 | 0 | 3 | 7 |
| 2 | 1 | 3 | 7 |
| 2 | 1 | 2 | 7 |
| 2 | 0 | 3 | 6 |
| 3 | 1 | 3 | 6 |
| 3 | 0 | 2 | 6 |
| 3 | 1 | 4 | 5 |
| 4 | 0 | 2 | 5 |
| 4 | 0 | 4 | 4 |
| 4 | 1 | 4 | 4 |
| 5 | 0 | 4 | 3 |
| 5 | 1 | 5 | 3 |
| 3 | 0 | 6 | 3 |
| 6 | 1 | 6 | 2 |
| 6 | 0 | 5 | 2 |
| 6 | 1 | 6 | 1 |

Table. 3-Dataset Marking Scheme

We have trained our model using above dataset. The values that we have defined in the yable are set according to the requirement of the answer. The evalautor/moderator/teacher of the answersheet can define these values for themselves to suit their needs.

# 4 EXPERIMENTAL ANALYSIS

We have given 3 questions to each student. Total number of student was 20.Each question carries 5 Marks .All answers are evaluated firstly by 10 Professors then our algorithm will evaluate them .Then the similarity betweenProfessorEvaluation and our algorithmevaluation is taken into consideration .we have found :-

| Outputs | Questions | Professor | NB Classifier |
|---|---|---|---|
| Student 1 | Q1 | 3 | 2.5 |
| | Q2 | 2 | 1.5 |
| | Q3 | 2 | 2.0 |
| Student 2 | Q1 | 1 | 1.5 |
| | Q2 | 1.5 | 1.5 |
| | Q3 | 2.5 | 2.0 |
| Student 3 | Q1 | 2 | 1.5 |
| | Q2 | 1 | 1.5 |
| | Q3 | 1.5 | 1.5 |
| .<br>.<br>.<br>Student 10 | | | |

Table. 4- Comparative Evaluation Result Table

We have made python flask web app for experiment purpose, where students will write the subjective question answersand we also have made an android application to show the results.


Fig. 2- Students Mark Evaluation Application Screenshot

# 5 CONCLUSION

The techniques discussed and implemented in this project should have a high agreement (up to 90 percent) with Human Performance. The project works with the same factors which an actual human being considers while evaluation such as length of the answer, presence of keywords, and context of key-words. Use of Natural Language Processing coupled with robust classification techniques, checks for not only keywords but also the question specific things. Students will have certain degree of freedom while writing the answer as the system checks for the presence of keywords, synonyms, right word context and coverage of all concepts. It is concluded that using ML techniques will give satisfactory results due to holistic evaluation. The accuracy of the evaluation can be increased by feeding it a huge and accurate training dataset. As the technicality of the subject matter changes different classifiers can be employed. Further improvement by taking feedback from all the stakeholders such as students and teachers can improve the system meticulously.

## References

[1] B. Rujiang and L. Junhua, Improving documents classification with semantic features, 2nd Int.Symp. Electron. Commer. Secur. ISECS 2009, vol. 1, pp. 640643, 2009.

[2] P. D. Turney and P. Pantel, From frequency to meaning: Vector space models of semantics, J.Artif. Intell. Res., vol. 37, pp. 141188, 2010.

[3] T. K. Landauer, Automatic Essay Assessment, Assess. Educ. Princ. Policy Pract., vol. 10, no. 3, pp.295308, 2003.

[4] T. K. Landauer, P. W. Foltz, and D. Laham, An introduction to latent semantic analysis, DiscourseProcess., vol. 25, no. 23, pp. 259284, 1998.

[5] T. K. Landauer and P. W. Foltz, An introduction to latent semantic analysis, Discourse Process.,no. April 2012, pp. 3741, 2012.

[6] P. W. Foltz, W. Kintsch, and T. K. Landauer, The measurement of textual coherence with latent semantic analysis, Discourse Process., vol. 25, no. 23, pp. 285307, 1998.

[7] P. W. Foltz, Latent Semantic Analysis for Text-Based, Behav. Res. Methods, Instruments Comput.,vol. 28, no. 2, pp. 197202, 1996.

[8] T. Kakkonen, N. Myller, E. Sutinen, and J. Timonen, Comparison of dimension reduction methodsfor automated essay grading, Educ. Technol. Soc., vol. 11, no. 3, pp. 275288, 2008.

[9] D. M. Blei, A. Y. Ng, and M. I. Jordan, Latent Dirichlet Allocation, J. Mach. Learn. Res., vol. 3,no. 45, pp. 9931022, 2012.

[10] T. Hofmann, Probabilistic latent semantic indexing, Sigir, pp. 5057, 1999.

[11] M. Islam, Automated Essay Scoring Using Generalized, in Proceesings of 13th InternationalConference on Computer and Information Technology (ICCIT 2010), 2010.

[12] L. Rudner and T. Liang, Automated essay scoring using Bayes theorem, J. Technol. Learn. ,vol. 1, no. 2, 2002.

[13] L. Bin, L. Jun, Y. Jian-Min, and Z. Qiao-Ming, Automated essay scoring using the KNN algorithm,Proc. - Int. Conf. Comput. Sci. Softw. Eng. CSSE 2008, vol. 1, pp. 735738, 2008.

[14] C. Leacock and M. Chodorow, C-rater: Automated scoring of short-answer questions, Comput.Hum., vol. 37, no. 4, pp. 389405, 2003.

[15] J. Z. Sukkarieh, Using a MaxEnt classifier for the automatic content scoring of free-text responses,AIP Conf. Proc., vol. 1305, pp. 4148, 2010.

[16] J. Sukkarieh and S. Stoyanchev, Automating Model Building in c-rater, Proc. 2009 Work. , no.August, pp. 6169, 2009.

[17] J. Burstein, K. Kukich, S. Wolff, C. Lu, M. Chodorow, L. Braden-Harder, and M. D. Harris, Automated scoring using a hybrid feature identification technique, Proc. 17th Int. Conf. Comput.Linguist. -, vol. 1, p. 206, 1998.

[18] D. Callear, J. Jerrams-Smith, and V. Soh, Bridging gaps in computerised assessment of texts, Proc.- IEEE Int. Conf. Adv. Learn. Technol. ICALT 2001, pp. 139140, 2001.

[19] P. Diana, A. Gliozzo, C. Strapparava, E. Alfonseca, P. Rodr, and B. Magnini, Automatic Assessment of Students free-text Answers underpinned by the Combination of a B LEU inspired algorithm and Latent Semantic Analysis, Mach. Transl., 2005.

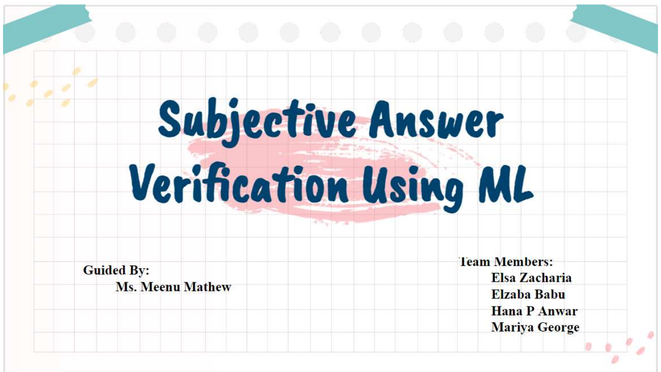
[20] F. Noorbehbahani and a. a. Kardan, The automatic assessment of free text answers using a modified BLEU algorithm, Comput. Educ., vol. 56, no. 2, pp. 337345, 2011.

[21] W. Wang and B. Yu, Text categorization based on combination of modified back propagation neural network and latent semantic analysis, Neural Comput. Appl., vol. 18, no. 8, pp. 875881, 2009.

[22] C. A. Kumar, M. Radvansky, and J. Annapurna, Analysis of a Vector Space Model , Latent Semantic Indexing and Formal Concept Analysis for Information Retrieval, vol. 12, no. 1, pp. 3448, 2012.

# Appendix B

# Presentation Slides


Subjective Answer Verification Using ML
Guided By:
Ms. Meenu Mathew
Team Members:
Elsa Zacharia
Elzaba Babu
Hana P Anwar
Mariya George

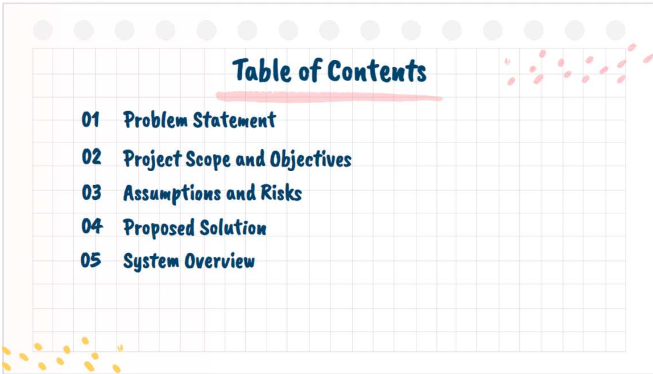


Table of Contents
01 Problem Statement
02 Project Scope and Objectives
03 Assumptions and Risks
04 Proposed Solution
05 System Overview

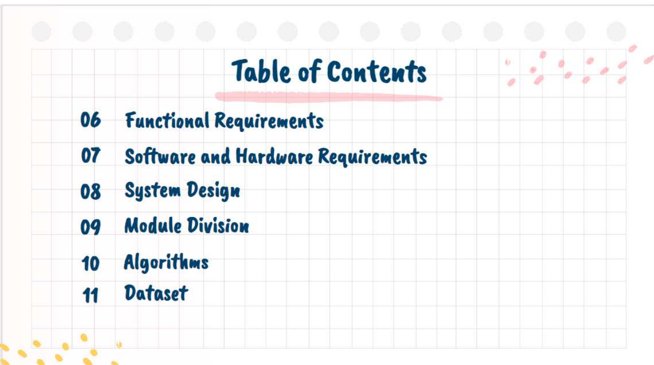


Table of Contents
06 Functional Requirements
07 Software and Hardware Requirements
08 System Design
09 Module Division
10 Algorithms
11 Dataset

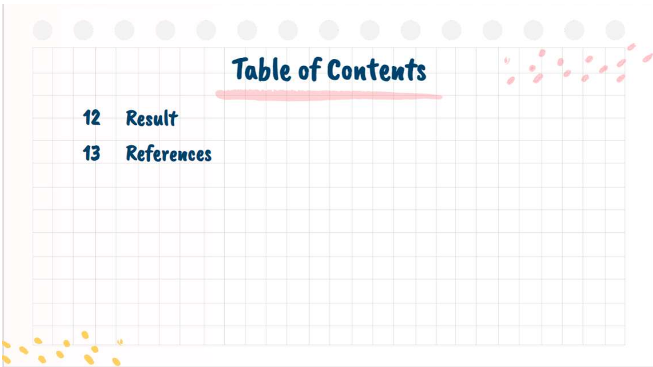


Table of Contents
12 Result
13 References

## 01
## Problem Statement

**Subjective questions are capable of examining the adopting ability of knowledge of the student, however, correctly evaluating the answer scripts is a challenging and complex job to perform, hence an efficient mechanism is needed to tackle this challenge by not only omitting human errors but also providing a quicker and faster output.**

## 02
## Scope And Objectives

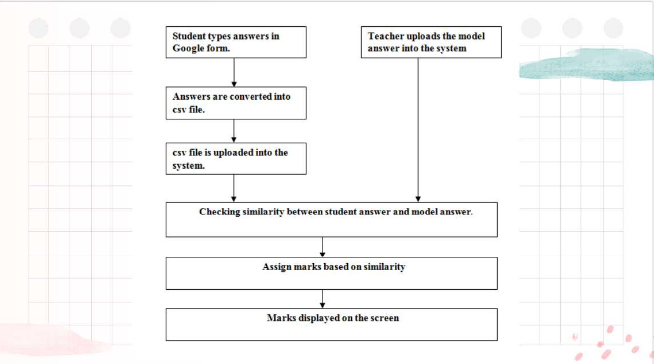
- The goal of this project is to develop a software to ease the burden on examiners.
- The answers will be typed by the students and various preprocessing steps using NLP techniques will be applied.
- Model answer sets will be provided by the evaluator.
- Compare the user answer with the model answer sets based on the keywords and grammar.
- Answers will be graded and results will be displayed accordingly.

## 03
## Assumptions And Constraints

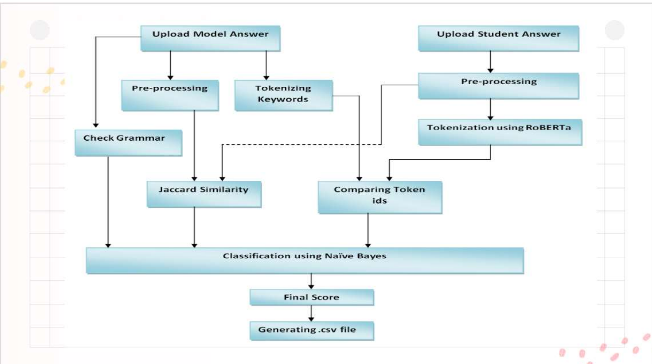
- System is used for evaluating only short subjective answers.
- Answers scripts are supposed to be in digital format.
- Algorithm will evaluate only theoretical answers and give marks according to the keywords and grammar.
- The answers consisting of non-textual data like equations,diagrams and tables will not be processed through this system.

- Difference in the answer structure of student answer and model answer.
- There is enormous ambiguity that exists while processing natural language.

## 04
## Proposed Solution




## 05
## System Overview




## 06
## Functional Requirements

1. Students enter answers into the google form.
2. Student answers are combined together into a csv file.
3. Teacher writes the model answers and corresponding keywords into a csv file.
4. Student answers are compared with the model answer.
5. Final scores are predicted and the results are generated and it is converted into a csv file.

## Hardware Requirements

**Processor**
Intel core i5 or i7

**Clockspeed**
1.8 GHz or faster

**RAM**
Minimum 8 GB

**Hard Disk Space**
50 GB or more

## Software Requirements

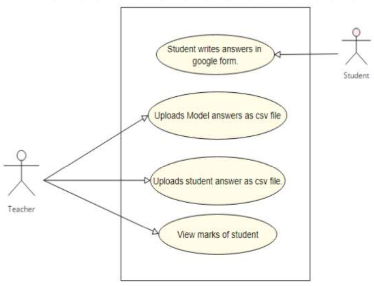
**Operating System**
Windows 7 or later

**Python**
Python 3 or later
Anaconda

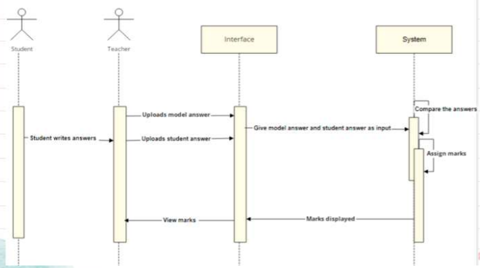
**Others**
Google Colab

# 08
# System Design

## Use Case Diagram




## Sequence Diagram



# 09

## Module Division

**Module 1-Preprocessing**

- Converting the text to lower case
- Removing special characters using regular expression
- Removing punctuations
- Removing white spaces

**Module 2-Tokenization using ROBERTA**

- Tokenization is done using RoBERTa Model(Robustly Optimized Pretraining Approach)
- It is based on Google's Bert model(2018)
- It is a pretrained model
- Same architecture as Bert model

**Module 3-Grammar Checking**

- This is done using Language tool Python.
- It is an open source grammar tool
- It allows you to detect grammar errors and spelling mistakes through a python script

**Module 4-QST Evaluation**

- QST is evaluated using Jaccard similarity.
- Jaccard similarity is used to check the similarity between two documents.
- Text distance,a python library compares the distance between 2 documents.

**Module 5-Classification**

- Naive Bayes model is used to classify into specific labels.
- These labels are used to predict the score.
- This is based on Bayes Theorem.

**Module 6-GUI**

- Tkinter is used to create GUI
- It is the standard library for python.
- Text alignments can be easily done using Tkinter.

# 10

## Modules

**1. GRAMMAR CHECKING**

```
def checkgrammer(text):

    # get the matches
    matches = tool.check(text)
    errors=len(matches)
    # print((errors))
    return errors
```

**2.TOKENIZATION**

```
def tokenfunction(inp):
    text=tokenizer.tokenize(inp,add prefix space=True)
    token ids = tokenizer.convert tokens to ids(text)
    print(token ids)
    return token ids
```

### 3. CLEANING

```
def cleantext(text):
  x=str(text).lower().replace('\\','').replace(' ','')
  x=x.replace('(','').replace(')','')
  #print(x)
  x=x.replace('"','').replace('.',' ')

  tag=' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)"," ",x).split())
  spcl=tag.replace('[^\w\s]','')
  return spcl
```

### 4. QST EVALUATION

```
        token1=str(clean ans).split()
        tokens2 = str(user clean ans).split()
        print("user_clean_ans text==>",user clean ans)
        QST=int(textdistance.jaccard(token1 , tokens2)*10)
```

# 11

# DATASET

- Training data is created as a list of 3 combinations- Keywords,Grammar,QST.
- There are a total of 62 such combinations.
- These combinations are mapped into different class labels.
- Class labels are in the range 0-10.

# 12

# RESULT

| | |
|---|---|
| accuracy | |
| macro avg | 0.79 |
| weighted avg | 0.86 |

# 13

# Literature Survey

**Piyush Patil,Sachin Patil,Vaibhav Miniyar and Amol Bandal,"Subjective Answer Evaluation Using Machine Learning",International Journal of Pure and Applied Mathematics,Volume 118,No.24,2018**

| AUTHORS | METHODS USED | ADVANTAGES | DISADVANTAGES | YEAR |
|---|---|---|---|---|
| Piyush Patil,Sachin Patil,Vaibhav Miniyar,Amol Bandal | This uses NLP techniques along with TF-IDF vectoriser for word embedding. | • Easy to compute similarity between two documents.<br>• Easy way to extract the most descriptive keywords in a document. | • Cannot capture semantics.<br>• May assign low values to words that are relatively important. | 2018 |

**Jagadamba G and Chaya Shree G,"Online Subjective answer verifying system Using Artificial Intelligence",Proceedings of the Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC),IEEE Xplore Part Number:CFP20OSV-ART; ISBN: 978-1-7281-5464-0,2020**

| AUTHORS | METHODS USED | ADVANTAGES | DISADVANTAGES | YEAR |
|---|---|---|---|---|
| Jagadamba G, Chaya Shree G | This method uses AI based answer verifier along with various NLP preprocessing techniques and Cosine similarity for sentence matching | • Robust due to allocation of appropriate weights to answers.<br>• Highly efficient for evaluating extensive subjective answers. | • An Interface that consists of Login credentials are required to type the answers.<br>• The proposed system is efficient only if the answer size is large enough. | 2020 |

# 14

# REFERENCES


[1] Shreya Singh,Omkar Manchekar,,Ambar Patwardhan,Prof. Uday Rote,Prof. Sheetal Jagtap,Dr. Hariram Chavan,"Tool for Evaluating Subjective Answers using AI(TESA)"IEEE International Conference on Communication information and Computing Technology (ICCICT), June 25-27, 2021, Mumbai, India

[2] Piyush Patil,Sachin Patil,Vaibhav Miniyar and Amol Bandal,"Subjective Answer Evaluation Using Machine Learning",International Journal of Pure and Applied Mathematics,Volume 118,No.24,2018

[3] Neethu George, Sijimol PJ and Surekha Mariam Varghese,"Grading Descriptive Answer Scripts Using Deep Learning",International Journal of Innovative Technology and Exploring Engineering (IJITEE),ISSN: 2278-3075, Volume-8 Issue-5 March, 2019

[4] Jagadamba G and Chaya Shree G,"Online Subjective answer verifying system Using Artificial Intelligence",Proceedings of the Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC),IEEE Xplore Part Number:CFP20OSV-ART; ISBN: 978-1-7281-5464-0,2020

# Appendix C
# Program Code

## GUI.py

```
from tkinter import *
import time
import re
from PIL import Image
#Import scikit-learn metrics module for accuracy calculation
import pickle
from PIL import Image, ImageTk
from tkinter import messagebox
from tkinter.filedialog import askopenfile
import numpy as np
from transformers import RobertaModel, RobertaTokenizer
import pandas as pd
import re
import textdistance
import language_tool_python
tool = language_tool_python.LanguageTool('en-US')
from datetime import datetime
tokenizer = RobertaTokenizer.from_pretrained('roberta-base', truncation=True,
do_lower_case=True)

from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
train_data=[[10,0,10],[10,0,9],[9,0,10],[9,1,10],[8,0,9],[8,1,9],[7,0,8],[7,1,8],[6,0,7],[
6,1,7],[6,0,6],[6,1,6],[6,0,5],[6,1,5],[6,0,4],[6,1,4],[6,0,3],[6,1,3],[6,0,2],[5,0,7],[5,1,7
],[5,0,6],[5,1,6],[5,0,5],[5,1,5],[5,0,4],[5,1,4],[5,0,3],[5,1,3],[4,0,7],[4,0,5],[4,1,6],[4,1
,5],[4,0,4],[4,1,4],[4,0,3],[4,1,3],[4,1,2],[3,0,4],[3,1,4],[3,0,3],[3,1,3],[3,0,2],[3,1,2],[2
,0,5],[2,0,3],[2,1,4],[2,1,3],[2,0,2],[2,1,2],[2,0,1],[2,1,1],[1,0,2],[1,1,2],[1,0,0],[1,0,1],
[1,1,1],[1,1,0],[0,0,1],[0,1,1],[0,1,0],[0,0,0]]
train_labels=[10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,
10,10,10,10,10,10,10,10,9,9,9,8,8,8,8,7,7,7,7,6,6,6,6,5,5,5,5,4,4,4,4,3,3,3,2,2,1,0,0]
print(len(train_data))
print(len(train_labels))
```

```
traindata=np.array(train_data)
trainlabel=np.array(train_labels)

X_train, X_test, y_train, y_test = train_test_split(traindata, trainlabel, test_size =
0.20, random_state = 45)

classifier = GaussianNB()
classifier.fit(X_train, y_train)

print("Model loaded.....")

def checkgrammer(text):

    # get the matches
    matches = tool.check(text)
    errors=len(matches)
    # print((errors))
    return errors

def tokenfunction(inp):
    text=tokenizer.tokenize(inp,add_prefix_space=True)
    token_ids = tokenizer.convert_tokens_to_ids(text)
    print(token_ids)
    return token_ids

def cleantext(text):
  x=str(text).lower().replace('\\','').replace('_','')
  x=x.replace('(','').replace(')','')
  #print(x)
  x=x.replace('"','').replace('.',' ')

  tag=' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)"," ",x).split())
  spcl=tag.replace('[^\w\s]','')
  return spcl
```

```
def pp(a):
    global mylist
    mylist.insert(END, a)

def predict(modelpath,answerpath):
    global mylist

    modeldf=pd.read_csv(modelpath)
    totalqstns=len(modeldf)
    print(totalqstns)
    stdf=pd.read_csv(answerpath)

    datacol=[]
   datacol.append("Student")
for i in range(totalqstns):
    datacol.append("Question "+str(i+1))
datacol.append("total")
print(datacol)
datalist=[]
stnamelist=[]
for i in range(totalqstns):
    qid=modeldf["Qid"][i]
    model_answer=modeldf["Answers"][i]
    keyw=modeldf["Keywords"][i]
    givenkeywords=str(keyw).split(",")
    givenvectors=[]
    scorelist=[]
    for key in givenkeywords:
        tkn=tokenfunction(key)
        for tk in tkn:
            givenvectors.append(tk)
    print("Model answer :: ",model_answer)
    clean_ans=cleantext(model_answer)
    for j in range(len(stdf)):
        stname=stdf["Name"][j]
        if(stname not in stnamelist):
            stnamelist.append(stname)
```

```
    userans=stdf.iloc[:,i+1][j]
    print("User answer :: ",userans)
    usergrammer=checkgrammer(userans)
    user_clean_ans=cleantext(userans)
    keywords=tokenfunction(user_clean_ans)
    keylist=[]
    for gk in givenvectors:
        for uk in keywords:
            if(gk==uk):
                if uk not in keylist:
                    keylist.append(uk)
    token1=str(clean_ans).split()
    tokens2 = str(user_clean_ans).split()
    print("user_clean_ans text==>",user_clean_ans)
    QST=int(textdistance.jaccard(token1 , tokens2)*10)
    Kwrd=int(len(keylist)/len(givenvectors)*10)
    if(usergrammer>3):
        grerror=1
    else:
        grerror=0
    print(Kwrd)
    print(grerror)
    print(QST)

    nblist=[Kwrd,grerror,QST]
    testdata=np.array(nblist)
    y_pred  =  classifier.predict(testdata.reshape(1,-1))
    print("ypred",y_pred)
    scorelist.append(y_pred[0])
    print("======================")
datalist.append(scorelist)


print(datalist)
print(stnamelist)
mainlist=[]
```

```
    for i in range(len(stnamelist)):
        sublist=[]
        sublist.append(stnamelist[i])

        tot=0
        for k in datalist:
            sublist.append(k[i])
            tot=tot+int(k[i])
        sublist.append(tot)
        mainlist.append(sublist)
    print("result==>",mainlist)


    now = datetime.now()
    dt_string = now.strftime("%d-%m-%Y")
    df = pd. DataFrame(mainlist, columns = datacol)
    df.to_csv(dt_string+".csv")
    print(df)
    messagebox.showinfo("Info", "Result saved to your directory")
    root.after(500, lambda : pp("Model answer uploaded"))
    root.after(2000, lambda : pp("Students answer uploaded"))
    root.after(2300, lambda : pp("Model Loaded"))
    root.after(2500, lambda : pp("Evaluating score"))
    root.after(2500, lambda : pp("Result saved as "+dt_string+".csv"))
    # root.after(2800, lambda : pp("Result: "+prediction))
    root.after(3000, lambda : pp("============================="))




def browseim():
    global cimg,shrslt,E1
    path = askopenfile()
    n=path.name
    print(n)
    E1.delete(0,"end")
    E1.insert(0, n)
```

```
def browseim2():
    global cimg,shrslt,E2
    path = askopenfile()
    n=path.name
    print(n)
    E2.delete(0,"end")
    E2.insert(0, n)

def userHome():
    global root, mylist,shrslt,E1,E2
    root = Tk()
    root.geometry("1200x700+0+0")
    root.title("Home Page")



    image = Image.open("back.jpg")
    image = image.resize((1200, 700), Image.ANTIALIAS)
    pic = ImageTk.PhotoImage(image)
    lbl_reg=Label(root,image=pic,anchor=CENTER)
    lbl_reg.place(x=0,y=0)

    #-----------------INFO TOP------------
    lblinfo = Label(root, font=( 'aria' ,20, 'bold' ),text="Subjective answer
Evaluation",fg="white",bg="#000955",bd=10,anchor='w')
    lblinfo.place(x=450,y=50)

    lblinfo3 = Label(root, font=( 'aria' ,20 ),text="Upload model answer
file",fg="#000955",anchor='w')
    lblinfo3.place(x=120,y=220)
    E1 = Entry(root,width=30,font="veranda 20")
    E1.place(x=50,y=260)
    btntrn=Button(root,padx=10,pady=2, bd=4 ,fg="white",font=('ariel'
,16,'bold'),width=10, text="Browse", bg="red",command=lambda:browseim())
    btntrn.place(x=180, y=300)
```

```
    lblinfo4 = Label(root, font=( 'aria' ,20 ),text="Upload student answer
file",fg="#000955",anchor='w')
    lblinfo4.place(x=120,y=380)
    E2 = Entry(root,width=30,font="veranda 20")
    E2.place(x=50,y=420)

    btn3=Button(root,padx=10,pady=2, bd=4 ,fg="white",font=('ariel'
,16,'bold'),width=10, text="Browse", bg="red",command=lambda:browseim2())
    btn3.place(x=180, y=460)

    mylist = Listbox(root,width=60, height=20,bg="white")

    mylist.place( x = 700, y = 200 )

    btnhlp=Button(root,padx=80,pady=8, bd=6 ,fg="white",font=('ariel'
,10,'bold'),width=7, text="Evaluate",
bg="blue",command=lambda:predict(E1.get(),E2.get()))
    btnhlp.place(x=150, y=550)


    def qexit():
        root.destroy()

    root.mainloop()

userHome()
```

# Appendix D
# Mapping the Project Objectives with POs and PSOs

# Course Outcome

| Sl No. | Description | Blooms Taxonomy Level |
|---|---|---|
| CS492.1 | Think innovatively on the development of components, products, processes or technologies in the engineering field. | Knowledge(Level 1)<br>Analyse(level 4) |
| CS492.2 | Apply knowledge gained in solving real life engineering problems . | Evaluate(level 2)<br>Understand(level 5) |

# CO-PO Mapping

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CS492.1 | - | 3 | - | 2 | - | - | 1 | - | 3 | 2 | 1 | 3 |
| CS492.2 | 3 | 2 | 3 | 2 | 2 | - | - | 2 | 3 | 2 | - | 1 |

# CO-PSO Mapping

| | PO1 | PO2 | PO3 |
|---|---|---|---|
| CS492.1 | 3 | - | 1 |
| CS492.2 | 3 | 3 | 2 |

# Justifications for CO-PO/PSO Mapping

| Mapping | Low/Medium/High | Justification |
|---|---|---|
| CS492.1–PO4 | M | Conduct investigations of complex problems : I used research based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. |
| CS492.1–PO7 | L | Environment and sustainability : I understood the impact of the professional engineering solutions in societal and environmental contexts, and demonstrated the knowledge of and the need for sustainable developments. |
| CS492.1–PO9 | H | Individual: We were able to function effectively as an individual, in multidisciplinary settings. |
| CS492.1–PO10 | M | Communication : We were able to communicate ef- fectively on complex Engineering activities with the Engineering Community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions. |
| CS492.1–PO11 | L | Project Management and finance : Demonstrated knowledge and understanding of the Engineering and management principles and apply these to one's own work, to manage projects and in multidisciplinary environments. |
| CS492.1–PO12 | H | Life-long learning : Recognized the need for, and have the preparation and ability to engage in inde- pendent and life-long learning in the broadest con- text of technological change. |

| | | |
|---|---|---|
| CS492.1–PSO1 | H | Computer Science Specific Skills : Was able to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science. |
| CS492.1–PSO3 | L | Professional Skills : Was able to apply the fundamentals of computer science to formulate competitive research proposals and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur. |
| CS492.2–PO1 | H | Engineering Knowledge : Applied the knowledge of Mathematics, Science, Engineering fundamentals, and an Engineering discipline to the solution of complex engineering problems. |
| CS492.2–PO2 | M | Problem analysis : We were able to identify, formulate, review research literature, and analyze complex Engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and Engineering sciences. |
| CS492.2–PO3 | H | Design/Development of solutions : Designed solu- tions for complex Engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal and environmental considerations. |
| CS492.2–PO4 | M | Conduct investigations of complex problems : Used research based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. |

| | | |
|---|---|---|
| CS492.2–PO5 | L | Modern Tool usage : Created, selected, and applied appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex Engineering activities with an understanding of the limitations. |
| CS492.2–PO8 | M | Ethics : Applied ethical principles and committed to professional ethics and responsibilities and norms of the Engineering practice. |
| CS492.2–PO9 | H | Individual: We were able to function effectively as an individual, and in multi-disciplinary settings. |
| CS492.2–PO10 | M | Communication : Communicated effectively on complex Engineering activities with the Engineering Community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions. |
| CS492.2–PO12 | L | Life-long learning : Recognized the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change. |
| CS492.2–PSO1 | H | Computer Science Specific Skills : We were able to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science. |
| CS492.2–PSO2 | H | Programming and Software Development Skills : Acquired programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products. |

| | | |
|---|---|---|
| CS492.2–PSO3 | M | Professional Skills : Applied the fundamentals of computer science to formulate competitive research proposals and to develop innovative products to meet societal needs thereby evolving as an eminent researcher and entrepreneur. |