INTELLIGENT ANSWER SCRIPT EVALUATION USING MACHINE LEARNING

A PROJECT REPORT

Submitted by

SATHYA.R (312316104155)

VIOLA GRACE VINITHA.P (312316104191)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

St. JOSEPH'S COLLEGE OF ENGINEERING, KANCHEEPURAM

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2020

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**Intelligent answer script evaluation using machine learning**" is the bonafide work of "Sathya .R" and "Viola Grace Vinitha .P", who carried out the project work under my supervision.

Dr. A. Chandrasekar M.E., Ph.D.,
HEAD OF THE DEPARTMENT
Professor
Department of Computer Science and Engineering,
St.Joseph's College of Engineering,
Old Mamallapuram Road,
Chennai-600119

Dr.D.Rosy Salomi Victoria,M.E.,M.S.,Ph.D.
SUPERVISOR
Associate Professor,
Department of Computer Science and Engineering,
St.Joseph's College of Engineering,
Old Mamallapuram Road,
Chennai-600119

CERTIFICATE OF EVALUATION

COLLEGE NAME: St. Joseph's College of Engineering, Chennai- 600 119

BRANCH : B.E (Computer Science Engineering)

SEMESTER : VII

SL.NO	NAME OF THE STUDENTS	TITLE OF THE PROJECT	NAME OF THE SUPERVISOR WITH DESIGNATION
1.	SATHYA.R	**INTELLIGENT ANSWER SCRIPT EVALUATION USING MACHINE LEARNING**	Dr. D.Rosy Salomi Victoria,M.E.,M.S.,Ph.D, Associate Professor.
2.	VIOLA GRACE VINITHA.P		

The report of the project work submitted by the above students in partial fulfilment for the award of Bachelor of Engineering Degree in Computer Science and Engineering of Anna University was confirmed to be a report of the work done by the above students and then evaluated.

Submitted to Project and Viva Examination held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

At the outset we would like to express my sincere gratitude to our beloved **Chairman, Dr.Babu Manoharan, M.A., M.B.A., Ph.D.,** for his constant guidance and support.

We would like to express our heartfelt thanks to our respected **Managing Director Mrs. S. Jessie Priya, M.Com.,** for her kind encouragement and blessings.

We wish to express our sincere thanks to **Director Mr. B. Shashi Sekar , M.Sc.,** for providing ample facilities in the institution.

We express our deepest gratitude and thanks to our beloved **Principal Dr.Vaddi Seshagiri Rao, M.E., M.B.A., Ph.D., F.I.E.,** for his inspirational ideas during the course of the project.

We wish to express our sincere thanks and gratitude to **Dr. A. Chandrasekar, M.E., Ph.D.,** Head of the Department of Computer Science and Engineering and my project guide **Dr. D. Rosy Salomi Victoria, M.E.,M.S.,Ph.D.,** Associate Professor, Department of Computer Science and Engineering, St.Joseph's College of Engineering for their guidance and assistance in solving the various intricacies involved in the project.

Our special thanks to Project coordinator **Mrs. P.N Jeipratha, M.E.,(ph.D.),** Assistant Professor, Department of Computer Science and Engineering for the deluge of ideas and assistance she has provided to us all through this project.

Finally we thank our parents and friends who helped us in the successful completion of this project.

ABSTRACT

Education is fundamental for human progress. A student is evaluated by the mark he/she scores. The evaluation of student’s work is a central aspect of the teaching profession that can affect students in significant ways. Though teachers use multiple criteria for assessing student work, it is not known if emotions are a factor in their grading decisions. Also, there are several mistakes that occur on the department's side like totalling error and marking mistake. So, we are developing software using Natural Language Processing. There are mainly two modules where Tika parser can be used to convert handwritten font to digital text and evaluation module evaluates the answer based on various factors and assign marks. For every answer being entered is evaluated based on the keywords and grammatical meaning of the sentence. With this approach we can save the cost of checking the answers manually and reduce the workload of the teachers by automating the manual checking process. The evaluation time is also reduced by using this software.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACRONYM	EXPANSION
CBOW	Continuous Bag-Of-Words
CLI	Command-Line Interface
CSS	Cascading Style Sheet
CSV	Comma-Separated Values
HTML	HyperText Markup Language
IDE	Integrated Development Environment
JPEG	Joint Photographic Experts Group
ML	Machine Learning
MLP	Multi-Layer Perception
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
NN	Neural Network
OCR	Optical Character Recognition
PDF	Portable Document Format
PIL	Python Imaging Library

RDBMS	Relational DataBase Management System
SQL	Structured Query Language
TF-IDF	Term Frequency-Inverse Document Frequency
UI	User Interface
URL	Uniform Resource Locator

CHAPTER 1

INTRODUCTION

1.1 PROBLEM IDENTIFICATION

Education is important for every human. A student is evaluated by the marks he/she scored. Now, a student writes his/her answer on the paper and the paper is evaluated by the human. But manually evaluating large amounts of handwritten data is a difficult process that's bound to be fraught with errors. It takes lots of time for evaluation and includes various human errors like totalling error, correction error. Manual evaluation also involves various human emotions also. There is another method for evaluation introduced in Bangalore, Karnataka called on-screen Evaluation. In this approach, after an examination, the paper-based answer scripts are sent to a secure scanning centre. The student/candidate details are masked, and bar-codes are assigned for tracking. The answer scripts are scanned, digitized, encrypted, and stored in secure data centres. The answer is electrically displayed on the screen, the teacher will check whether the keyword presents in the answer or not. Based on this, they allocate the marks. In this also human involves so there may be chances of errors. So we introduce a navel approach, where there is no human involved.

It includes there modules, handwritten to text conversion, answer evaluation and UI to display marks. The scanned answer image will be given as input to the first module, it extracts text from the image .This text given as input to the second module, it evaluates answer based on various factor and calculated mark. The mark is displayed on the web page which is user interface.

1.2 PROJECT OBJECTIVE

The main objectives of this work:

1. To create image to text conversion module, through this module we extract text from the image, which is given as input. This module gives extracted text as a text document.

2. To create Evaluation module, which gets text file, which is the answer script as an input. This module separates answers based on question number. It evaluates the answer based on various factors like grammar, number of sentences and number of keywords match. Mark is generated for each question. Thereby the total mark is generated.
3. To create a User Interface to upload answer script and to display the mark obtained by the student for the particular subject.
4. To create a database design to store the generated mark with the student's detail.

1.3 SIGNIFICANCE OF WORK

Software deals with converting handwritten font to text using Neural Network (NN) based Handwritten Character Recognition system i.e. OCR with parsing and evaluating the answer scripts using evaluation algorithm. Optical Character Recognition involves the detection of handwritten text on images and conversion of the handwritten text to encoded text that the computer can easily understand. Then the encoded text will be given as input to the evaluation Model. The basic grammar and spellings, stop words removal, lemmatization and WordNet generation are done in answer. Answers are checked with keywords and the mark is calculated. Thus the system is trained with dataset to evaluate the answer scripts. With the fed keyword set, the system would evaluate any answer script, thus the technology involved is Machine Learning (ML). It is hoped that this work could provide significant impact on the evaluation time. Also, it is hoped that this software will reduce various errors like totalling error, marking error etc. because there is no human involved in evaluation.

TABLE 1.1. EXISTING SYSTEM VERSUS PROPOSED SYSTEM

EXISTING SYSTEM	PROPOSED SYSTEM
Involves human interaction for evaluating the answer script	Completely automated
The possibility of emotions being reflected in their correction.	No such suspicion.

More human resources are needed.	Less human resource.
Lengthy process.	Takes less time.
Logistical cost and travel management of evaluators is high.	Logistical cost and travel management of evaluators is eliminated.
Physically answer sheet to be stored in a central location insecure environment.	No need for storage of answer sheets.
There are a timeline and pressure to complete the entire answer sheet evaluation activity within the stipulated timeline.	The proposed system can help to simplify the result generation process

1.4 OVERVIEW OF PROJECT

Chapter 1: Introduction

This chapter introduces background information and highlights the main issues that drive the work. The main objectives of the work are presented which includes the significance of the project work.

Chapter 2: Literature Review

The chapter summarizes the literature survey which has been conducted. It contains coverage of the main established concepts and techniques published in the literature concerning subjective answer script evaluation. A short summary of image to text conversion and evaluation process characteristics is also presented in this chapter for clearer understanding of the subject. A survey of the existing results for different methods to evaluate answers is also highlighted. This chapter concludes by providing a basis or motivation for continuation of the project.

Chapter 3: Process Description

This chapter describes how Optical Character Recognition extracts text from image .It also describes how the evaluation algorithm evaluates the answer and allocates marks. The generated mark is stored in the database.

Chapter 4: System Design

The hardware and software requirements are stated here.

Chapter 5: Extracting text from image

This chapter describes converting handwritten font to text using Neural Network based Handwritten Character Recognition system. Also it describes Segmentation where the text is broken down into characters. This chapter concludes by explaining how the software is able to match the characters through comparison and various detection algorithms.

Chapter 6: Answer evaluation

This chapter explains identification of grammar mistakes using grammar check algorithm. Parsing is done. Then it gives full description about the removal of stop words from the answer. It also describes lemmatization of the sentence where all the words in the sentence are converted into their root form. It also explains semantic and lexical keyword matching. Finally it concludes by calculation of mark.

Chapter 7: User Interface Design

This chapter describes the displaying of marks on the web page which is the User Interface (UI).It also explains how the answer image, keyword file and total marks uploaded in the UI. Then it concludes with displaying marks on the web page. The marks along with student details is stored in the database is also displayed.

Chapter 8: Database Design

This chapter describes storing student details, subject details and database details in Database. MySQL database is used for database .because we are going to store only structured data like string, integer, float and etc,. These values will be fetched from webpage and stored in database. This details are later used to display whole student results.

Chapter 9: Result and Discussion

This chapter summarizes the difference between manual evaluation marks and software evaluation marks. And also it explains reasons for more accurate result from the software.

Chapter 10: Conclusion and Future Enhancement

This chapter summarizes the major conclusions of the work and suggests the further work to be carried out in the future. Various sources pertaining to this work are listed in the references.

CHAPTER 2

LITERATURE REVIEW

There are several algorithms that have been proposed for handwriting recognition and conversion.

Sardar Jaf, and Calum Calder [15] presented a paper on multi-lingual dependency parser. Using advanced deep learning techniques, our parser architecture tackles common issues with parsing such as long-distance head attachment. It implements a parser based on this architecture to utilize transfer learning techniques to address important issues related with limited-resourced language. This paper presents promising results for solving core problems in natural language parsing.

The paper presented by Rishi Verma1, Chris A. Mattmann [14] focuses on techniques to merge the parallelized data processing capabilities of Apache Spark with the extensive file-type parsing support of Apache Tika. The Tika parser is used here for making consistent information extraction of over 1,200 text and binary file types on a sequential file basis. In this proposal (1) analysis of numerous (1000x) un-partitioned small to medium sized Tika parse-able files, and (2) analysis of very large partition- able Tika parse-able files is done.

Evaluation of subject answer checking isn't a new thought. It has been in the works for a decade and a half. A large number of techniques were experimented with to solve the problem efficiently. Natural Language processing(NLP), Latent Semantic Analysis, Generalized Latent Semantic Analysis, Bayes theorem, K- nearest neighbour, etc. In general, they can be categorized as follows : Clustering techniques, classification techniques and natural language processing techniques.

Ms. Kranti Vithal Ghag, Dr. Ketan Shah [7] has proposed an analysis on sentiment classification model. It explains the Sentiment Classification model with a Preprocessing Techniques to Remove Stopwords. Stopwords such as "the", "and", "is' were removed. Natural Language Tool Kit (NLTK) English Corpus Stopwords list was used.

V.SenthilKumaran, A.Sankar [19] developed a model that takes short answers as input and constructs RDF sentences. Using an ontology student answer is mapped with a model answer they confirmed the same keyword is not needed to get full marks. Even the synonym of the keyword also gets marks. For One-Sentence answer, matching with the model answer is done by taking into consideration both lexical structure and synonyms. However, there is a restriction imposed on the length of the sentences of the answer.

The paper by Jin Xu,Yubo Tao,Hai [5] Lin deals with Semantic Word Cloud.Semantic word cloud is taking into account for the word semantic meanings. Distributed word representation is applied to accurately describe the semantic meaning of words, and a word similarity graph is constructed based on the semantic distance between words to lay out words in a more compact and aesthetic manner. It used to explore the theme in the text.

The importance of key words is measured by the Term Frequency-Inverse Document Frequency (TF-IDF) value. To create similarity graph Node set and Edge set used. Node set is important keywords and edge will be created if the similarity between two words satisfies the user defined threshold. The semantic meanings of words are provided by word embeddings in this paper, which is first used in the semantic word cloud generation to the best of our knowledge. We prepare the related text corpus and then train our word embeddings by using the Continuous Bag-Of-Words (CBOW) model, which is implemented in the open-source toolkit word2vec.

CHAPTER 3

PROCESS DESCRIPTION

This project deals with converting handwritten font to text using Tika parser, a Neural Network based Optical Character Recognition system with parsing and evaluating the answer scripts. Optical Character Recognition involves the detection of handwritten text on images and conversion of the handwritten text to encoded text that the computer can easily understand. The areas containing text can now be broken down into lines and words and characters and the software is able to match the characters through comparison. Then the encoded text will be given as input to the evaluation Model. The basic grammar and spellings are checked using a grammar check model. Through this model, the percentage of correctness words can be found. Stop words are removed and lemmatization is done. Answers are checked with keywords, which will be given as a dataset. The words with similar meanings may occur in the answer sheet, so we also generate WordNet to accept the words and ignore reducing marks. Then the total mark is calculated based on the correctness of the sentence and number of keywords matched. The total mark along with student's details is stored in the database for future reference.

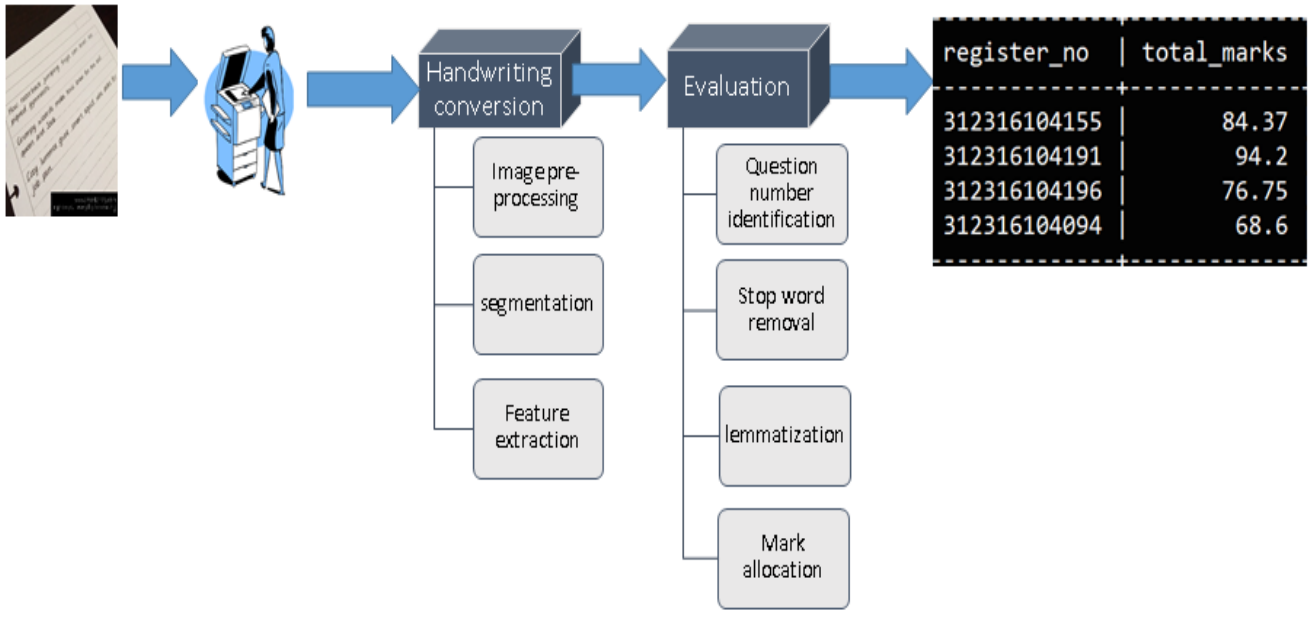


Figure 3.1 ARCHITECTURE DIAGRAM

Figure 3.1 illustrates the process flow. The answer sheet from the student is scanned and handwritten text is converted to machine encoded text, the converted text is evaluated, marks are calculated and stored in the database.

CHAPTER 4

SYSTEM DESIGN

In this chapter we present technologies used to develop a project. And also here we explains software requirement and hardware requirements.

i. Machine Learning

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task. It would be very hard to perform OCR with a rules based approach, but with machine learning it is much easier and more reliable. OCR is almost always implemented with machine learning. The technology allows replicating human ability to recognize various text patterns, fonts, or styles on images, documents, video streams, etc. It becomes possible due to the capability of machines to learn. OCR algorithms work on convolutional neural networks of different types. Such algorithms are usually trained on some input datasets. Some core functions are to localize image files, improve the quality, detect and recognize characters, and turn files into machine-readable formats.

ii. Natural Language Processing

Natural Language Processing (NLP) is a subfield of linguistics, computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of natural language data. It is used to apply machine learning algorithms to text and speech. Due to NLP, computers can interact with written (as well as spoken) forms of human language.

4.1 Hardware Requirements

a) High speed scanners
b) Processor -I3,I5I7
c) RAM -4GB
d) Hard Disk -250GB

a) High Speed Scanners

High Speed non-destructive scanning is needed to convert handwritten answer scripts to machine encoded text. High-speed scanners scan the answer sheets with the proper masking process. Student identity information like name, Roll No is masked. Examiner would not be able to see these details. It would enable a transparent and fair evaluation process. It is possible to scan the bar-code of the answer sheet and map it with specific student records.

b) Processors

Generally speaking, Core i7s are better than Core i5s, which are in turn better than Core i3s. Core i7 does not have seven cores nor does Core i3 have three cores. The numbers are simply indicative of their relative processing powers.

c) RAM

If you wish to enhance the performance, as well as, the graphics output of your system, you can do so by adding 4 GB of RAM, which is more than enough to ensure that you can multitask without a hitch.

d) HARD DISK

A 250 GB hard disk is the minimum requirement to run the software.

4.2 Software Requirements

a) Python version 3.6 and above
b) PyCharm community software or Jupyter Notebook
c) MySQL
d) Python packages
    - nltk

- Pandas
- Tika
- Flask
- Mysql
- Os

a) **Python**

Python's expansive library of open source data analysis tools, web frameworks, and testing instruments make its ecosystem one of the largest out of any programming community. Python is an accessible language for new programmers because the community provides many introductory resources.

Python is widely used for analyzing the data but the data need not be in the required format always. In such cases, we convert that format (like PDF or JPEG etc.) to the text format, in order to analyze the data in better way. Python offers many libraries to do this task.
Python programs that work with human language data for applying in statistical Natural Language Processing (NLP) are run through a platform, Natural Language Toolkit (NLTK).

b) PyCharm community software or Jupyter Notebook

PyCharm is an IDE which means it's used for writing real code. Notebooks are for interactive settings. They are an improvement over the python shell because it makes it easier to rerun commands and save graphics in one place. PyCharm makes it easier for programmers to write various web applications in Python supporting widely used web technologies like HTML, CSS, JavaScript.

We may also use Jupyter Notebook because Jupyter supports various programming languages. Jupyter makes it easier to distribute research results, as notebooks are easy to share with others. Jupyter Notebooks can also be used as a user interface for big data frameworks, databases and computer clusters and it is open source.

c) Mysql

We use MySQL because it is one of the best RDBMS being used for developing various web-based software applications. It is a fast, easy-to-use RDBMS being used for many small and big businesses. It is released under an open-source license. It also works very quickly and works well even with large data sets.

d) Python Libraries and packages

- **Natural Language Tool Kit (NLTK)**

  This is a suite of libraries and programs for symbolic and statistical NLP for English. It ships with graphical demonstrations and sample data. We will perform tasks like NLTK tokenize, removing stop words, stemming NLTK, lemmatization NLTK and more. We import stopwords, RegexpTokenizer, PorterStemmer, wordnet, WordNetLemmatizer.

- **Pandas**

  Pandas is an open source Python package that provides numerous tools for data analysis. The package comes with several data structures that can be used for many different data manipulation tasks. It also has a variety of methods that can be invoked for data analysis, which comes in handy when working on data science and machine learning problems in Python. Pandas has a variety of utilities to perform Input / Output operations in a seamless manner. We use it to read data from a variety of formats such as CSV, MS Excel, etc.

- **Tika**

  Apache Tika is a library that is used for document type detection and content extraction from various file formats. Tika uses existing various document parsers and document type detection techniques to detect and extract data. Tika provides a single generic API for parsing different file formats. It uses existing specialized parser libraries for each document type. We use Tika Parser to extract text from scanned answer script i.e. pdf.

- **Flask Web Framework**

  We use Flask to render the webpage. We will also use the Flask web framework to create our simple OCR server where we can take pictures via the webcam or upload photos for character recognition purposes.

- **MySQL**

  We connect to the database, create an object for the database, execute the SQL query, fetch records from the result and inform the Database if any changes are made to the table.

- **OS**

  Python OS module allows us to use the operating system dependent functionalities and to interact with the underlying operating system in several different ways. We can work with files, change the environment variables, and we can move files around, etc. This is as same as overriding all the OS built-in functionalities in a module and using them in a file I/O and system handling. We use this to read the file from the system in which the software runs.

CHAPTER 5

IMAGE TO TEXT CONVERSION (OCR)

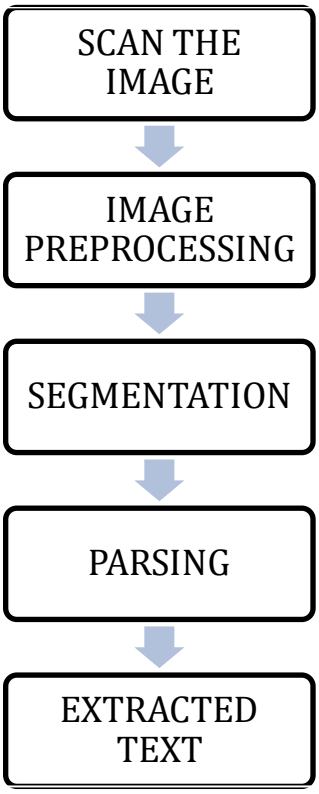


Figure 5.1 Handwriting Conversion Process Flow

5.1 Image pre-processing

Pre-processing is a common name for operations with images at the lowest level of abstraction both input and output are intensity images. The aim of pre-processing is an improvement of the image data that suppresses unwanted distortions or enhances some image features important for further processing. Image pre-processing is important in the recognition pipeline for correct character prediction and conversion. Image pre-processing methods typically include noise removal, image segmentation, cropping, scaling, and more. The character recognition system first accepts a scanned image as an input. The images can be in JPG or PNG or JPEG format. Image pre-processing methods use the considerable redundancy in images. Neighbouring pixels corresponding to one object in real images have essentially the same or similar brightness value. Thus, distorted pixel can often be restored as an average value of neighbouring pixels. Four categories of image pre-processing methods

according to the size of the pixel neighbourhood that is used for the calculation of a new pixel brightness:

- pixel brightness transformations
- geometric transformations

Pre-processing methods that use a local neighbourhood of the processed pixel. Image restoration that requires knowledge about the entire image. Other classifications of image pre-processing methods exist.

Digital capture and conversion of an image often introduces noise, which makes it hard to identify the actual text. Noise means, the pixels in the image show different intensity values instead of true pixel values that are obtained from image. Noise removal algorithm is the process of removing or reducing the noise from the image. The noise removal algorithms reduce or remove the visibility of noise by smoothing the entire image leaving areas near contrast boundaries. The common types of noise that arises in the image are:

Impulse noise, Additive noise, Multiplicative noise.

Different noises have their own characteristics which make them distinguishable from others. Therefore it is necessary to reduce noise, while preserving the strokes of the characters, since they are important for correct classification of text.

5.2. Segmentation

Segmentation of hand written text document into individual character or digit is an important phase in document analysis, character recognition and many other areas. Character segmentation has become a crucial step for mail address recognition in the automatic post mail sorting system. Also out of available text segmentation methods, we do not have a universal accepted solution. The reason for not achieving satisfactory recognition rates is the difficult nature of cursive handwriting and difficulties in the accurate segmentation and recognition of cursive and touching characters. In order to segment text from a given input document image, it is necessary to detect all the possible text regions. In the case of printed scripts, segmentation is a relatively simple task. In the case of overlapped scripts, broken characters, connected characters, loosely configured

characters, and mixed scripts, segmentation is difficult. Overlapped, broken, connected and loosely configured characters are major causes of segmentation errors. Segmentation of Text Image is used to locate each individual character and its boundaries. It involves process of labeling, which assigns the some label to spatially align units i.e., pixel, connected components or characteristic points such that a group of pixels with the similar label share specific visual features.

Segmentation is breaking the whole image into subparts to further process them. In this stage, a sequence of characters is partitioned into a sub-image of an individual character. Each character is resized into 30×20 pixels. Segmentation of characters is faster than any conventional method in which all the characters from the text are segmented by connected component processing only. Segmentation algorithm recognises line by line and word by word detection. Efficient algorithm is used for segmentation.

5.3 PARSING

Parsing is the process of analysing a string of symbols, either in natural language, computer languages or data structures, conforming to the rules of a formal grammar. Parsing is done "To split a file or other input into pieces of data that can be easily stored or manipulated." So we are splitting a string into parts then recognizing the parts to convert it into something simpler than a string.

CHAPTER 6

ANSWER EVALUATION

- The text document(Answer Script) is given as input to the evaluation module.
- The basic grammar and spellings are checked using grammar check model. Through this model percentage of correctness word can be found.
- Answers are checked for keywords, which will be given as dataset.
- Stop words are removed and keywords are matched.
- Marks are allotted based on the match and semantic check, i.e words with similar meanings are also considered.
- Total mark is calculated.

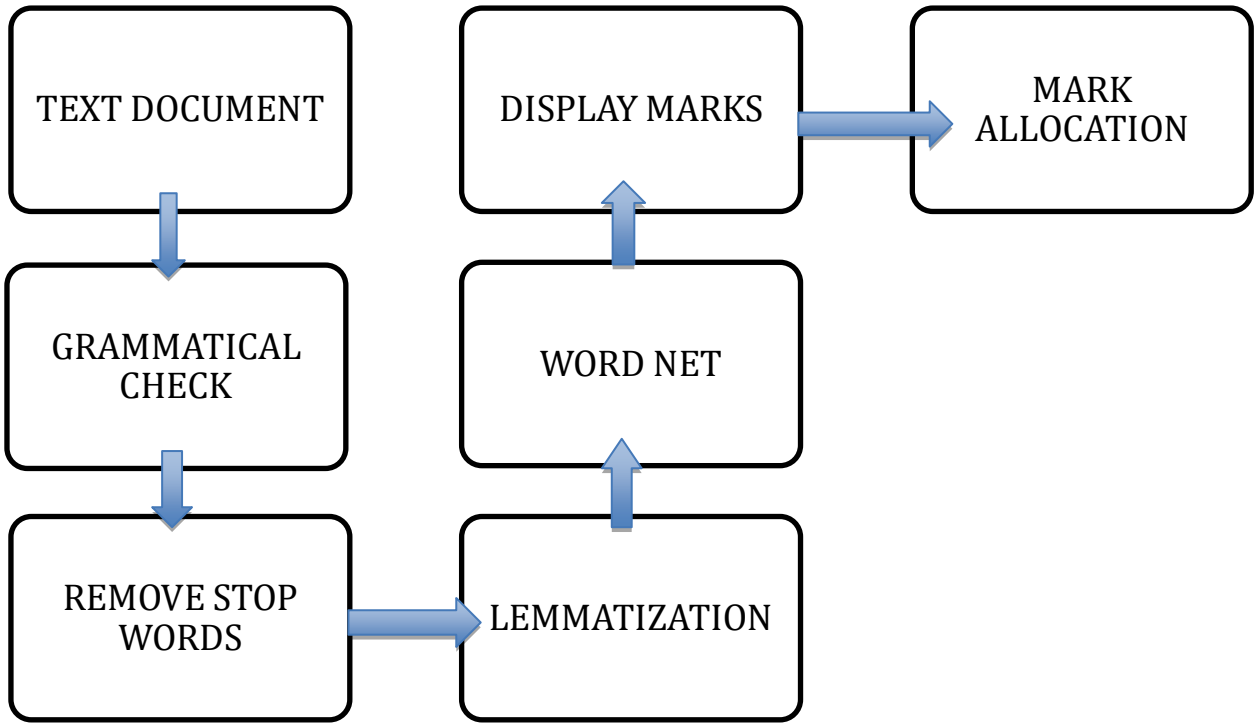


Figure 6.1 Evaluation Process Flow

6.1 Introduction

The text file which contains the answer is given as input to this module. This answer will undergo various steps like Grammar check, Stop words removal, stemming, lemmatization, wordnet. Finally, the mark will be calculated based on grammar correctness and keyword match.

6.2 Question number separation

Question number is identified and the sub-divisions like a, b are identified.

6.3 Grammar check

A grammar checker function will find each word in a sentence, look up each word in the dictionary, and then attempt to parse the word into a form that matches a grammar. Using various rules, the program can then detect various errors, such as tense, spelling mistakes, etc.

Number of mistakes in the text will be found. Using this percentage of correctness of the sentence will be calculated and then word with mistake will be changed to the correct word. Natural Language Toolkit (NLTK) has support for grammars that you can use to parse your sentence. You can define a grammar, or use one that is provided, along with a context-free parser. If the sentence parses, then it has valid grammar; if not, then it doesn't. The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical Natural Language Processing (NLP) for English written in the Python programming language. NLTK supports classification, tokenization, stemming, tagging, parsing, and semantic reasoning functionalities.

6.4 Remove stop words

Stopwords are the English words which does not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence. For example, the words like the, he, have etc**.** We would not want these words taking up space in our database, or taking up valuable processing time. For this, we can remove them easily, by storing a list of words that you consider to be stop words. NLTK in python has a list of stopwords stored in 16 different languages. Thus NLTK helps in removing stop words. The correct sentence from the Grammar check will be sent to stop words removing function. The function will remove

stop words like and, but, then, is, etc.. are from the correct sentence. It will return the array containing keywords only. This is how we are making our processed content more efficient by removing words that do not contribute to any future operations.

6.5 Lemmatization

Lemmatization is the process of grouping together the different inflected forms of a word so they can be analysed as a single item. Lemmatization is similar to stemming but it brings context to the words. So it links words with similar meaning to one word. To extract the proper lemma of word, it is necessary to look at the morphological analysis of each word. Rather than stemming, lemmatization provides lemma which has a dictionary meaning. The key to Lemmatization is linguistics.

6.6 WordNet

WordNet is a lexical database. The WordNet is a part of Python's Natural Language Toolkit. It is a large word database of English Nouns, Adjectives, Adverbs and Verbs. These are grouped into some set of cognitive synonyms, which are called synsets. It contains semantic relations between words in more than 200 languages. It links words into semantic relations including synonyms, meronyms. The synonyms are grouped into sets called synsets with short definitions and usage examples. Synonyms are words that have similar meanings. A synonym set, or synset, is a group of synonyms. A synset, therefore, corresponds to an abstract concept. Synsets form relations with other synsets to form a hierarchy of concepts, ranging from very general ("entity", "state") to moderately abstract ("animal") to very specific ("plankton"). Using Synsets, the meaning of the words in the answer and keywords are found. Check how many keywords are present in the answer by calculating distance. We can also easily use WordNet to compare the similarity of two words and their tenses. Given that synsets can be organized as a graph, as shown above, we can measure the similarity of synsets based on the shortest path between them. This is called the **path similarity**, and it is equal to 1 / (shortest_path_distance (synset1, synset2) + 1). It ranges from 0.0 (least similar) to 1.0 (identical).

6.7 Mark Calculation

Using the percentage of correctness of the sentence, number of keywords presented in the answer and length of the answer mark is calculated. Formula:

1. Mark Calculation for a Question= (Number of Keywords matched / Total Number of Keywords)*100
2. Total_marks = SUM(Each question’s Mark)

CHAPTER 7

USER INTERFACE DESIGN

UI design stands for “User Interface.” The user interface is the graphical layout of an application. The user interface, in the industrial design field of human-computer interaction, is the space where interactions between humans and machines occur. The admin has to upload the dataset for evaluation through web page. Our script can be used via the command line, but a Flask application would make it more user-friendly and versatile. For instance, we can upload photos via the website and get the extracted text displayed on the website or we can capture photos via the web camera and perform character recognition on them. The web page is created using the Python Flask Web framework to upload an answer image, keyword file and mark for the answer. Then click submit button to pass all these three as an argument to the function .The function will convert the answer image to text and evaluate the answer using keywords. Then the calculated mark will be displayed on the screen. Our Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. Flask application has been able to integrate the OCR functionality and display the mark on the browser. This makes it easier to process images instead of running commands on the CLI every time we have a new image to process.

Handling file upload in Flask is very easy. It needs an HTML form with its enctype attribute set to ‘multipart/form-data’, posting the file to a URL. The URL handler fetches file from request.files[] object and saves it to the desired location.Each uploaded file is first saved in a temporary location on the server, before it is actually saved to its ultimate location. Name of destination file can be hard-coded or can be obtained from filename property of request.files[file] object. However, it is recommended to obtain a secure version of it using the secure_filename() function.It is possible to define the path of default upload folder and maximum size of uploaded file in configuration settings of Flask object. The ‘/upload’ URL rule that displays ‘upload.html’ from the templates folder, and ‘/upload-file’ URL rule that calls uploader() function handling upload process.

Flask's main advantage is that it has access to a multitude of online resources for documentation purposes. It is one of the most used Python web frameworks, which is why there are available a lot of tutorials or libraries for it.

CHAPTER 8

DATABASE DESIGN

MySQL database is used to store student mark details. Database is created with the name ‘student’. This database contains one table named ‘details’. This table contains 6 fields called student_name, Register_number, subject_code, subject_name, marks and status. Once the mark is calculated it will stored in database. These details are fetched from HTML form.

Table 8.1: Database Design

```
mysql> desc details;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| student_name | varchar(50) | YES  |     | NULL    |       |
| register_no  | bigint(20)  | YES  |     | NULL    |       |
| subject_name | varchar(50) | YES  |     | NULL    |       |
| subject_code | varchar(10) | YES  |     | NULL    |       |
| total_marks  | float       | YES  |     | NULL    |       |
| status       | varchar(20) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```

This database can be used to store structured data like student details, subject details and mark details.

CHAPTER 9

RESULT AND DISCUSSION

The final result will be the mark for an answer. First the answer script will be scanned using high quality scanner. The scanned image will be given as input to first module. This module extracts text from image. Then, the text is evaluated using keywords. This is done in the second module. The mark is calculated using number of sentence in the answer, percentage of grammar correctness and number of keywords matched. Finally the mark will be displayed on the web page. Using this software, human interaction on the evaluation is completely removed. Unlike manual evaluation, Human is needed only to scan and upload the paper. So there is less error. This software avoids human error, time thereby being a more useful to the students and also to the centre of examination.

Table 9.1: Results for all students

serial_no	student_name	register_no	subject_name	subject_code	total_marks	status
1	sathya	312316104155	Software Project Management	MG6088	84.37	Best
2	viola grace vinitha	312316104191	Software Project Management	MG6088	94.20	Best
3	yogeswaran	312316104196	Software Project Management	MG6088	76.75	Average
4	mageshwari	312316104094	Software Project Management	MG6088	68.60	Average
5	Swakshan	312316104179	Software Project Management	MG6088	49.06	Worst
6	Swakshan	312316104179	Human Computer Interaction	CS6008	67.77	Average
7	Viola Grace Vinitha	312316104191	Human Computer Interaction	CS6008	82.45	Best
8	Sathya	312316104155	Human Computer Interaction	CS6008	44.17	Worst
9	Yogeswaran	312316104196	Human Computer Interaction	CS6008	76.72	Average
10	Mageshwari	312316104094	Human Computer Interaction	CS6008	31.76	Worst

Table 9.1 shows the marks of 10 students for two subjects, Software Project Management and Human computer interaction. These details will be retrieved from database and displayed in webpage.

Table 9.2: Mark allocation

Register No	Correct mark	Manual evaluation Mark	System Evaluation mark
312316104155	83	79	84.37
312316104191	94.5	95	94.20
312316104196	78	80	76.75
312316104094	68	65	68.60
312316104179	48	52	49.06

Table 9.2 contains marks by manual evaluation and machine evaluation. This table compares the marks awarded after human and machine evaluation. The first column depicts the register number. The second column depicts the correct marks. The third column depicts the mark awarded by human. The forth column depicts the mark awarded by the system.

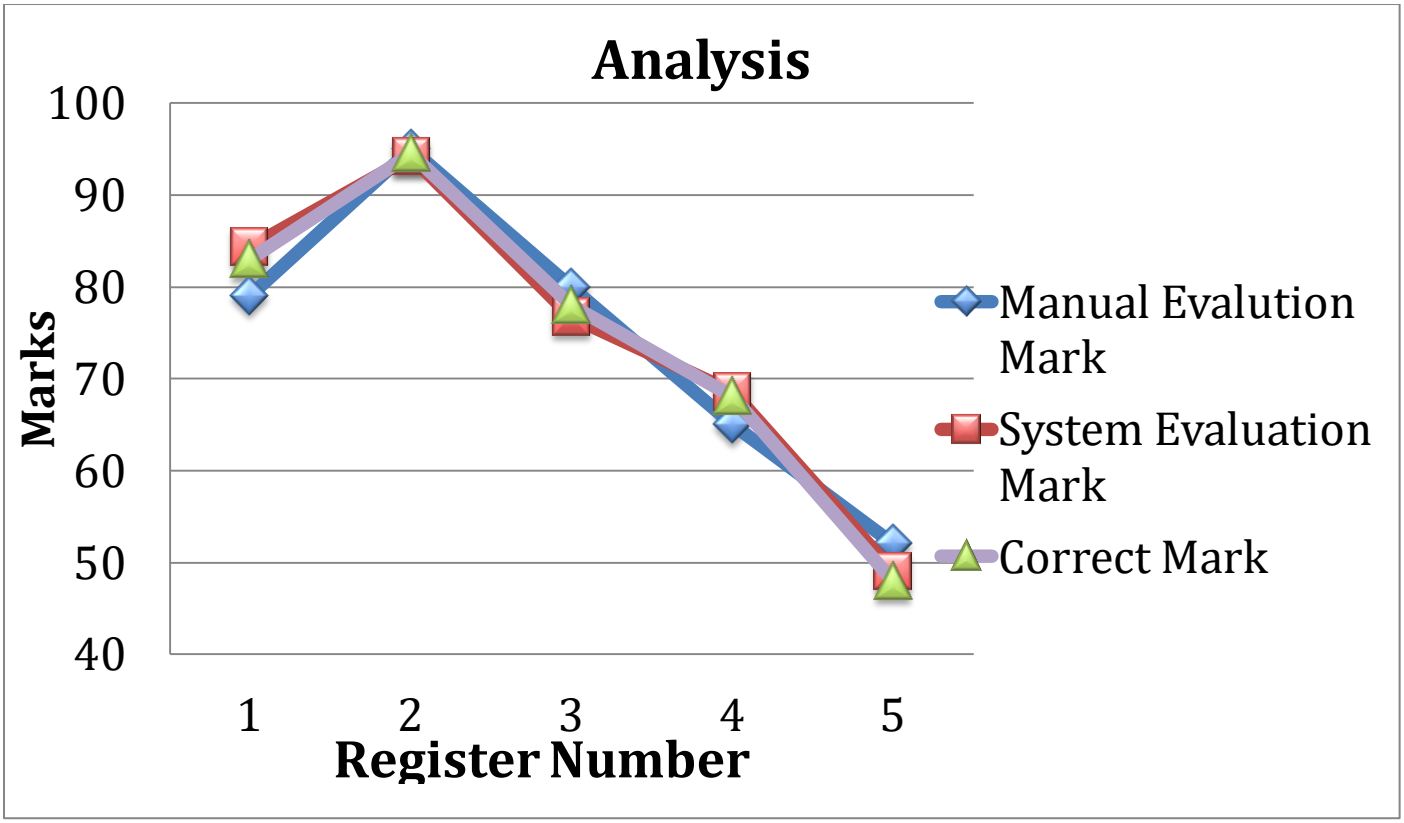


Figure 9.1: Line Chart For Comparison Of Normal Evaluation And System Evaluation

The above chart [Fig 9.1] is drawn using Table 9.1 of data. The Blue line shows the marks given by humans and the Red line shows marks given by software. For question 1, the correct mark will be 4.8, by human evaluation it gets 4.7 marks assigned, whereas by using a machine we can get an accurate answer 4.8. For question 4, the correct mark is 3.1, human evaluates this question and gives 4 marks, whereas machine evaluation gives 3 marks, which is closer to correct mark.

CHAPTER 10

CONCLUSION AND FUTURE ENHANCEMENTS

10.1 Conclusion

we have developed a model using an intelligent Character Recognition and Evaluation algorithm which evaluates theoretical answers and gives marks according to the keyword match. It reduces manual work and saves time with fast evaluation of answers and our model has produced good intermediate results. It was used to segment difficult cursive handwritten documents with some modifications and extract text from image.

10.2 Future Enhancement

The proposed system can be extended to work on degraded text or broken characters. Recognition of digits in the text, half characters and compound characters can be done to improve the word recognition accuracy. Font Independent OCR can be used to identify different font styles used by students/candidates.

The model can be trained in different languages. From this, we can create a dataset of different handwritten languages. Hence an answer with a non English language can also be evaluated. And the System will also evaluate the overwritten alphabets and other words with absolute accuracy.

The model can also be trained in such a way that it can check the complete paper instead of a single answer. Hence the model will evaluate the answers according to the answer number provided in the answer sheet. The model can be trained to evaluate diagrams and provide the marks accordingly. Thus the model with the user interface would need the user to upload the answer script which will be evaluated by the system and the marks will be stored in the database.

Evaluation algorithm produce marks as output by evaluating answers based on grammatical meaning of the sentence, length of the answer, number of important keywords matched.

APPENDIX I

SAMPLE CODING

```
#INTELLIGENT ANSWER SCRIPT EVALUATION USING MACHINE LEARNING

#EXTRACTION OF TEXT FROM IMAGE USING TIKA

def pdf_to_text(self,filename):
    raw = parser.from_file(filename)
    text=raw["content"]
    return text
#GRAMMAR CHECK

def grammar_check(self,filename):
    tool = language_check.LanguageTool('en-US')
    text = open(filename, "r")
    data = text.read()
    no_words = len(data.split())
    matches = tool.check(data)
    wrong = len(matches)
    return wrong,no_words


# GRAMMAR CORRECTION
def grammar_correction(self,data):
    tool = language_check.LanguageTool('en-US')
    matches = tool.check(data)
    corrected_data = language_check.correct(data, matches)
    return corrected_data


#STOP WORD REMOVAL
def stop_word_removal(self,data):
    tokenizer = RegexpTokenizer(r'\w+')
    stop_words = set(stopwords.words('english'))
```

```
    word_tokens = tokenizer.tokenize(data)
    filtered_sentence = [w for w in word_tokens if not w in stop_words]
    return (filtered_sentence)
#LEMMATIZATION

def lemmatization(self,data):
    porter_stemmer = PorterStemmer()
    lemme = []
    for w in data:
        lemme.append(porter_stemmer.stem(w))
    return lemme


#GENERATION OF WORDNET AND IDENTIFYING THE NUMBER OF MATCHED
#KEYWORDS

def wordNet(self,answer,keywords):
    lemmatizer = WordNetLemmatizer()
    extended_keywords = []
    extended_answer = []
    keywords_matched = 0
    for word in keywords:
        temp_list = []
        temp_list.extend(lemmatizer.lemmatize(word))
        for happy_lemma in wn.lemmas(word):
            temp_list.append(happy_lemma.name())
            for related_lemma in happy_lemma.derivationally_related_forms():
                temp_list.append(related_lemma.name())
        temp_list = list(set(temp_list))
        extended_keywords.append((word, temp_list))

    keywords_dictionary = {key: value for (key, value) in extended_keywords}
    answer_dictionary = {key: value for (key, value) in extended_answer}
```

```
    print("THE KEYWORD DICTIONARY")
    print(keywords_dictionary)
    print("THE ANSWER DICTIONARY")
    print(answer_dictionary)

    flg=0
    for keyword in keywords_dictionary.keys():
        flg=0
        for answer in answer_dictionary.keys():
            if keyword == answer:
                flg=1
                print("matched word:"+keyword+" "+answer)
                keywords_matched = keywords_matched + 1
                break

        if(flg==0):
            print("unmatched is:"+keyword)
    return keywords_matched


#SEPARATION OF QUESTION NUMBERS FROM ANSWERS

def question_no_split_ans(self,ans):
    words = ans.split(" ")
    mystr = ""
    iter = 1
    newstr = ''
    flg=0
    for i in range(0, len(words)):
        q_no = str(iter) + '.'
        words[i] = words[i].replace('\n', '')
        if(i!=(len(words)-1)):
```

```
        if (words[i] == q_no or words[i+1]== 'a.' or words[i+1]=='b.'):
            iter = iter + 1
            newstr = newstr + mystr
            mystr = "hai " + words[i] + " "
        else:
            mystr = mystr + words[i] + " "

    newstr = newstr + mystr
    newstr = newstr.split("BREAK ")
    return newstr

#SEPARATION OF SUB-DIVISIONS AND IDENTIFYING EXTRA OR DUPLICATE QUESTIONS
def ab_separation(self,ans,keyword):
    store='0'
    new_keyword_list = keyword[0:11]
    del keyword[:11]
    index=-1
    repetition= []
    for ans_iterate in ans:
        index=index+1#no of answers to know the index
        ans_iterate_word = ans_iterate.split(" ")
        if(store==ans_iterate_word[0]):
            repetition.append(index)
        for keyword_iterate in keyword:
            keyword_iterate_word = keyword_iterate.split(" ")
            if (ans_iterate_word[0] == keyword_iterate_word[0]):
                store=ans_iterate_word[0]
                if (ans_iterate_word[1] == keyword_iterate_word[1]):
                    new_keyword_list.append(keyword_iterate)
                    break
```

```
    print("REPEATED")
    print(repetition)
    return new_keyword_list,repetition


#DATABASE CONNECTIVITY

def database_connection(self, name, regno, sub_name, sub_code, tot, case):
    try:
        connection = mysql.connector.connect(host='localhost',
                                             database='student',
                                             user="root",
                                            password="viola")
        if connection.is_connected():
            db_Info = connection.get_server_info()
            print("Connected to MySQL Server version ", db_Info)
            mySql_insert_query = """INSERT INTO details
(student_name,register_no,subject_name,subject_code,total_marks,status)
                    VALUES
                    (%s,%s,%s,%s,%s,%s) """
            values = (name, regno, sub_name, sub_code, tot, case)
            cursor = connection.cursor()
            cursor.execute("select database();")
            record = cursor.fetchone()
            print("You're connected to database: ", record)
        cursor.execute(mySql_insert_query, values)
        connection.commit()
        print(cursor.rowcount, "Record inserted successfully into Details table")


    except Error as e:
        print("Error while connecting to MySQL", e)
    finally:
```

```
        if (connection.is_connected()):
            cursor.close()
            connection.close()
            print("MySQL connection is closed")

#READING THE UPLOADED FILE FROM THE SYSTEM

store_my_previous_mark = 0
if request.method=="POST":
    # total_marks=request.form['mark']
    # total_marks=int(total_marks)
    sub_name=request.form['sub_name']
    sub_code=request.form['sub_code']
    name=request.form['name_field']
    print(name)
    regno=request.form['regno_field']
    print(regno)
    mark=request.files['mark']

    file1= request.files['file_field']

    keyword=request.files['keyword']
    # print(keyword.filename)
    if file1.filename != '':
        ans_path=os.path.join(os.path.join(UPLOAD_DIRECTORY), file1.filename)
        file1.save(os.path.join(os.path.join(UPLOAD_DIRECTORY), file1.filename))
    if keyword.filename != '':
        key_path = os.path.join(os.path.join(UPLOAD_DIRECTORY), keyword.filename)
        keyword.save(os.path.join(os.path.join(UPLOAD_DIRECTORY), keyword.filename))
    if mark.filename != '':
```

```
    mark_path = os.path.join(os.path.join(UPLOAD_DIRECTORY), mark.filename)
    mark.save(os.path.join(os.path.join(UPLOAD_DIRECTORY), mark.filename))
```

```
#READING THE ANSWER SCRIPT, KEYWORD FILE, MARKS
ans=open(ans_path,"r")
anss=ans.read()
#----------------------------------here i add
answer=eval.question_no_split_ans(anss)#returned a list# answer=ans.read().split("\n\n")

print("THE RETURNED ANSWER VALUE IS")
print(answer)

keys = open(key_path, "r")
keyss=keys.read();
# ----------------------------------here i add#keyss = keys.read().split("\n\n")
keyword=eval.question_no_split_ans(keyss)#list returned

print("THE RETURNED KEYWORD VALUE IS")
print(keyword)
print(len(keyword))

mark = open(mark_path, "r")
marks = mark.read()
marks = eval.question_no_split_ans(marks)

#IF THE STUDENT HAS ATTENDED BOTH a and b FOR A QUESTION THEN THE
#ANSWER WHICH HAS HIGHEST MARK WILL BE CONSIDERED FOR TOTAL
```

```
#MARK CALCULATION
for data,keywords,mark in zip(answer,new_keyword_list,marks):
    count=count+1
    if data == '':
        continue
    elif keywords == '':
        continue
    else:
        # corrected_data = eval.grammar_correction(data)
        # corrected_keywords = eval.grammar_correction(keywords)
        print("--------------------------------------------------")
        print("\n\nTHE ANSWER IS\n")
        print(data)
        print("THE KEYWORDS ARE")
        print( keywords+"\n")
        print("THE TOTAL MARK ALLOTED IS")
        print("count",count)
        if(count>10):
            mark_al=mark[-3]+mark[-2]
            print(mark[-3]+mark[-2])
        else:
            mark_al=mark[-2]
        print("marks[-2]??????")
        print(mark_al)

        # print(corrected_keywords[1].split())
        stopword_removed_answer = eval.stop_word_removal(data)
        stopword_removed_keywords = eval.stop_word_removal(keywords)

        print("Stop word removed")
        print(stopword_removed_keywords)
```

```
lemme_answer = eval.lemmatization(stopword_removed_answer)
lemme_keywords = eval.lemmatization(stopword_removed_keywords)
print("lemme keywords")
print(lemme_keywords)
keywords_len=len(lemme_keywords)-1
print("keyword length")
print(keywords_len)

no_of_matched_words = eval.wordNet(lemme_answer, lemme_keywords)
# no_of_matched_words=3;
print("The number of matched words")
no_of_matched_words=no_of_matched_words-1
print(no_of_matched_words-1)

# percentage of keywords present in answer
match_percentage = (no_of_matched_words / keywords_len) * 100
print(match_percentage)
# overall_percentage=(grammar_percentage+match_percentage)/2

mark=float(mark_al)
print("----------MARK----------",mark)
q_mark = (match_percentage / 100) * mark
q_mark=round(q_mark, 2)

#what if 12a and 12b is written award highest mark

pop_it_out= 0
for i in repetition:
    if(i==count):
        print(store_my_previous_mark)
```

```
            if(store_my_previous_mark>q_mark):
                q_mark=store_my_previous_mark
            pop_it_out=1
        store_my_previous_mark=q_mark
        if(pop_it_out==1):
            print("I DELETED")
            del markss[-1]
            print(markss)
            pop_it_out=0
        markss.append(q_mark)
        mark_file = open("final_marks.xls", "w")
        n = mark_file.write(str(q_mark))
        n = mark_file.write("\n")
        mark_file.close()
        print("The mark is:\n", markss)


#IDENTIFYING THE BEST , WORST, AVERAGE CASES
tot=round(sum(markss),2)
if tot>=80:
    case="Best"
elif tot>=50:
    case="Average"
else:
    case="Worst"
print("The total mark is \n",tot)
```

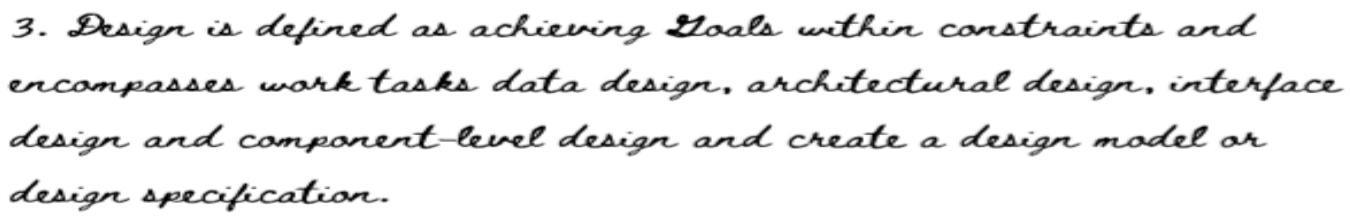
APPENDIX II

SCREENSHOTS

The project focuses on converting handwritten text to machine encoded text, evaluation of answer script and database storage. Different screenshots explain the different stages of the project and finally the output is generated.

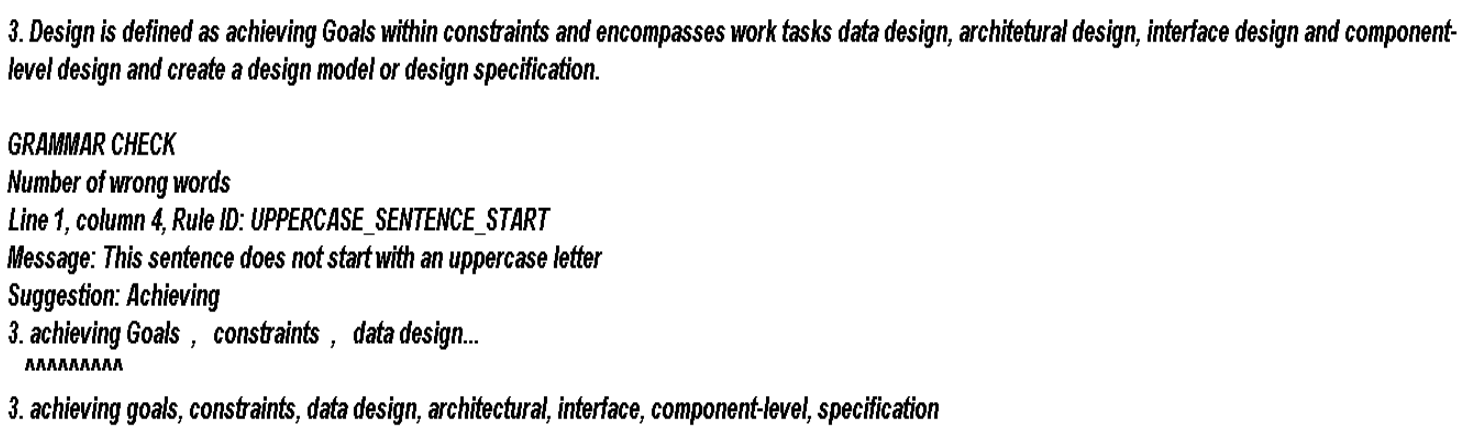
1. TEXT EXTRACTION

Input Image



Output Image:

2. COUNT OF MISTAKEN WORDS IN THE SENTENCE AND PREDICTED THE CORRECT WORD



3. **REMOVING STOP WORDS FROM A SENTENCE**

Stop word removed
['3', 'achieving', 'Goals', 'constraints', 'data', 'design', 'architectural',

'interface', 'component', 'level', 'specification']

4. LEMMATIZATION

lemme keywords
['3', 'achiev', 'goal', 'constraint', 'data', 'design', 'architectur', 'interfac', 'compon', 'level', 'specif']
keyword length
10

5. **KEYWORDS MATCH**

Number of keywords matched : 10

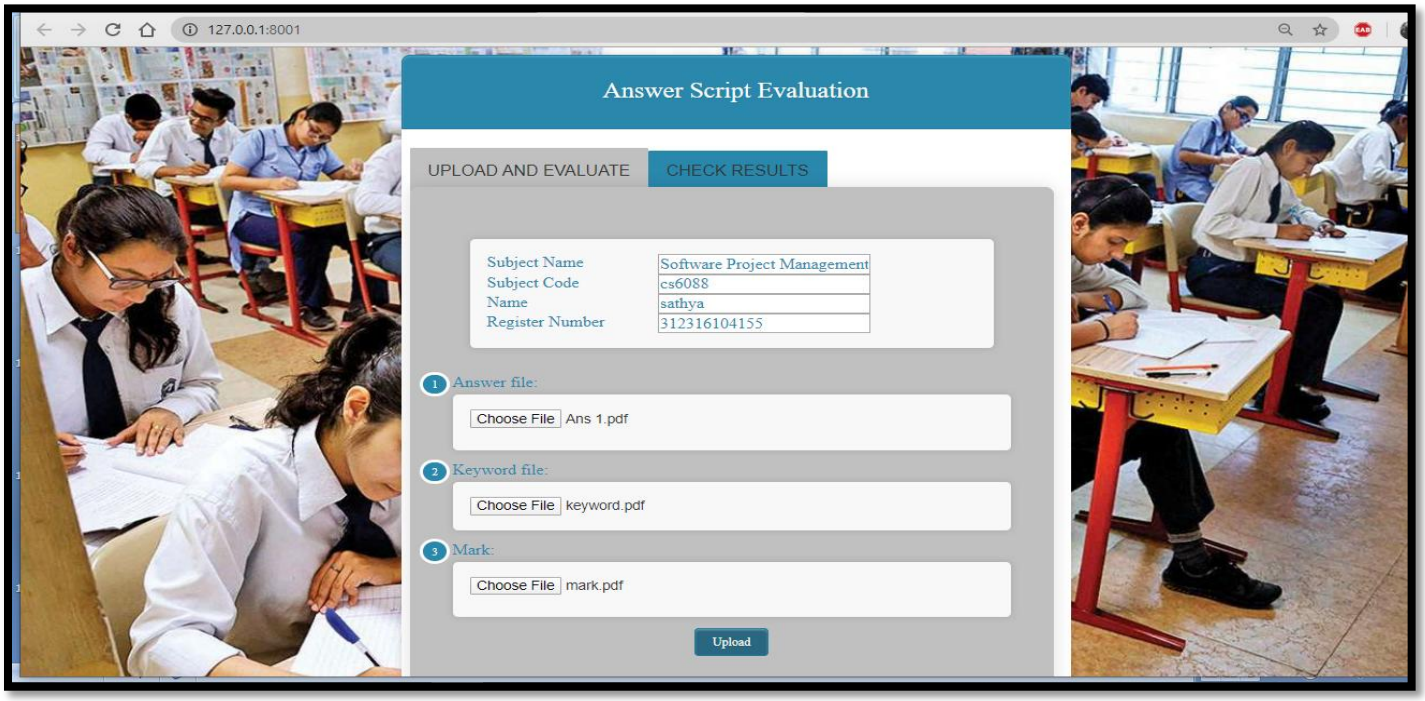
6. MARK GENERATED FOR A QUESTION

QUESTION NUMBER	MARKS
1	2.0

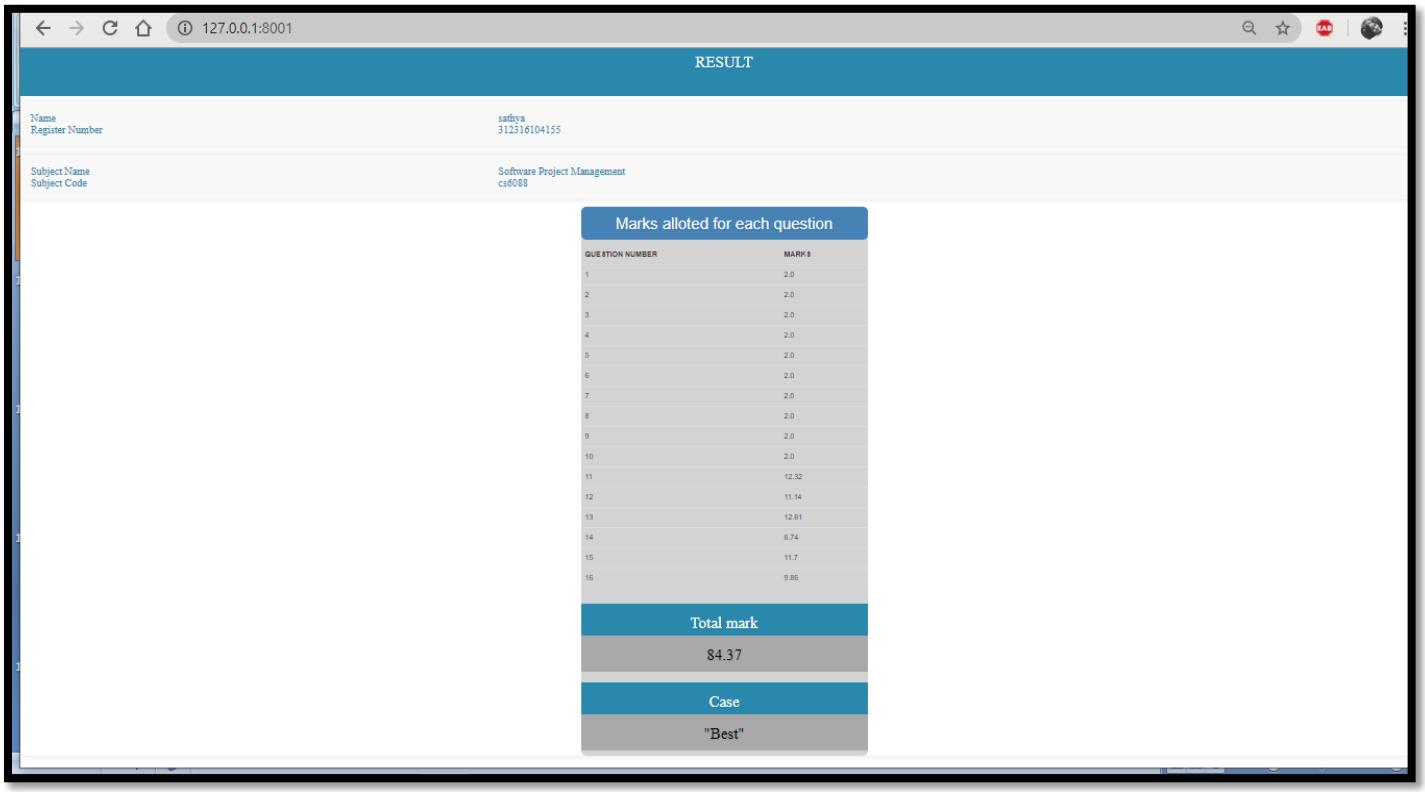
Total mark
84.37

7. OUTPUT SCREENS

The Answer Script along with Student's details is uploaded in this page.



Marks for each question that is calculated and displayed in this page.



Marks for each student that is calculated and displayed in this page

serial_no	student_name	register_no	subject_name	subject_code	total_marks	status
1	sathya	312316104155	Software Project Management	MG6088	84.37	Best
2	viola grace vinitha	312316104191	Software Project Management	MG6088	94.20	Best
3	yogeswaran	312316104196	Software Project Management	MG6088	76.75	Average
4	mageshwari	312316104094	Software Project Management	MG6088	68.60	Average
5	Swakshan	312316104179	Software Project Management	MG6088	49.06	Worst
6	Swakshan	312316104179	Human Computer Interaction	CS6008	67.77	Average
7	Viola Grace Vinitha	312316104191	Human Computer Interaction	CS6008	82.45	Best
8	Sathya	312316104155	Human Computer Interaction	CS6008	44.17	Worst
9	Yogeswaran	312316104196	Human Computer Interaction	CS6008	76.72	Average
10	Mageshwari	312316104094	Human Computer Interaction	CS6008	31.76	Worst

REFERENCES

1. Chhanda Roy and Chitrita Chaudhuri (2018) 'Case Based Modeling of Answer Points to Expedite Semi-Automated Evaluation of Subjective Papers', IEEE 8th International Advance Computing Conference (IACC), pp. 85-90.
2. Gaurav, K. and Bhatia P. K. (2013) 'Analytical Review of Preprocessing Techniques for Offline Handwritten Character Recognition', 2nd International Conference on Emerging Trends in Engineering & Management (ICETEM), pp. 14-22.
3. Jaideepsinh K. Raulji and Jatinderkumar R. Saini (2016) 'Stop-Word Removal Algorithm and its Implementation for Sanskrit Language', International Journal of Computer Applications (0975 – 8887),Vol. 150 ,No.2, pp. 15-17.
4. Jentrisi Priyatno and Moch Arif Bijaksana (2019) 'Clustering Synonym Sets in English WordNet', 7th International Conference on Information and Communication Technology (ICoICT), pp. 1-4.
5. Jin Xu, Yubo Tao and Hai Lin (2016) 'Semantic Word Cloud Generation Based on Word Embeddings', IEEE Pacific Visualization Symposium 2016 19–22 April, Taipei, Taiwan 978-1-5090-1451-4/16, pp. 239-243.
6. Kissan G. Gauns Dessai, Venkatesh, Kamat, V. and Ramrao S. Wagh (2014) 'Effective Use of Rubrics in Computer Assisted Subjective Answer-script Evaluation', IEEE Sixth International Conference on Technology for Education, pp .95-98.
7. Kranti Vithal Ghag and Ketan Shah (2015) 'Comparative Analysis of Effect of Stopwords Removal on Sentiment Classification', IEEE International Conference on Computer, Communication and Control (IC4-2015) .
8. Marina Nakano, Emi Masuda and Masaru Kamada (2016) 'A Structure Editor for the English Language', International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), pp. 155-161.
9. Mohammed Alshahrani, Spyridon Samothrakis and Maria Fasli (2017) 'Word mover's distance for affect detection', International Conference on the Frontiers and Advances in Data Science (FADS), pp.18-23.

10. Patil, S.B. (2011) 'Neural Network based bilingual OCR system: experiment with English and Kannada bilingual document', International Journal of Computer Applications, pp. 6-14.
11. Prayag Singh, Saurabh Sheorain, Shivam Tomar, Shubham Sharma and Bansode, N.K. (2018) 'Descriptive Answer Evaluation', International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Vol. 05 Issue: 05, pp.2709-2714 .
12. Priya S Nayak, Thejaswini, M.R., Kumar, N.S. (2018) 'Flexible Question Wise Evaluation of Digitized Answer Scripts', IEEE Ninth International Conference on Technology for Education (T4E), pp. 216-217.
13. Rahul Pramanik and Soumen Bag (2018) 'Shape decomposition-based handwritten compound character recognition for Bangla OCR', Journal of Visual Communication and Image Representation, Vol. 50, pp 123-134.
14. Rishi Verma1 and Chris A. Mattmann (2015) 'Extending Spark Analytics through Tika-based Information Extraction and Retrieval', IEEE 16th International Conference on Information Reuse and Integration, pp. 215-217.
15. Sardar Jaf and Calum Calder (2019) 'Deep Learning for Natural Language Parsing', IEEE Access Vol:07, pp.67-72.
16. Shahbaz Hassan, Ayesha Irfan, Ali Mirza and mran Siddiqi (2019) 'Cursive Handwritten Text Recognition using Bi-Directional LSTMs: A case study on Urdu Handwriting', International Conference on Deep Learning and Machine Learning in Emerging Application (Deep ML), pp. 67-72.
17. Subhadip Basu, Nibaran Das, Ram Sarkar and Mita Nasipuri (2012) 'An MLP based Approach for Recognition of Handwritten Bangla Numerals', Indian International Conference on Artificial Intelligence, pp. 407-417.
18. Suzen, N., Gorban, A.N. , Levesley, J. and Mirkes, E.M. (2018) 'Automatic Short Answer Grading and Feedback Using Text Mining Methods', arXiv:1807.10543v1 [cs.CL], pp.1-20.

19. SenthilKumaran, V. and Sankar, A. (2015) 'Towards Automated System For short-answer assessment using ontology mapping', International Arab Journal of e-Technology, Vol. 4, No. 1, pp. 17-24.
20. VenkateswaraRao, N. , Srikrishna, A. , RaveendraBabu, B. and Babu, G.R.M. (2011) 'An efficient feature extraction and classification of handwritten digits using neural networks', International Journal of Computer Science Engineering and Applications, pp. 47-56.
21. Xinhui Wu and Hui Li **(**2017) 'Topic Mover's Distance Based Document Classification', 17th IEEE International Conference on Communication Technology, pp. 1998-2002.