DEVELOPMENT OF COMPUTER BASED ANSWER SCRIPT EVALUATION SYSTEM

Md Nurul Amin

ID: ASH1811021

Session: 2017-18

Bachelor of Science in Information and Communication Engineering

Noakhali Science and Technology University

March 2023

DEVELOPMENT OF COMPUTER BASED ANSWER SCRIPT EVALUATION SYSTEM

Submitted by:

Md Nurul Amin

ID: ASH1811021

Session: 2017-18

Supervised by:

Dr. Md Ashikur Rahman Khan

Professor

Department of Information and Communication Engineering

Noakhali Science & Technology University

Bachelor of Science in Information and Communication Engineering

Noakhali Science and Technology University

March 2023

DEVELOPMENT OF COMPUTER BASED ANSWER SCRIPT EVALUATION SYSTEM

MD NURUL AMIN

This Research proposal is submitted in fulfillment of the requirements for the award of the degree of Bachelor of Science in Information and Communication Engineering

BACHELOR OF SCIENCE IN INFORMATION AND COMMUNICATION ENGINEERING

NOAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY

March 2023

SUPERVISOR'S DECLARATION

This thesis report is submitted to the Department of Information & Communication Engineering, Noakhali Science & Technology University, Sonapur, Noakhali in partial fulfillment of the requirements for having the B.Sc. degree in ICE.

_____

Prof. Dr. Md. Ashikur Rahman Khan

Professor

Department of Information and Communication Engineering

Noakhali Science & Technology University

STUDENT'S DECLARATION

I hereby declare that this thesis report has not been submitted elsewhere for the requirement of any kind of Degree, Diploma, or Publication.

Md Nurul Amin

Roll No: ASH1811021M

Session: 2017-2018

Dept of Information and Communication Engineering

Dedicated to my Parents

ACKNOWLEDGEMENTS

I am grateful and would like to express my sincere gratitude to my supervisor Professor Dr. Md. Ashikur Rahman Khan for his germinal ideas, invaluable guidance, continuous encouragement, and constant support in making this research possible. He has always impressed me with his outstanding professional conduct, his strong conviction for science, and his belief that a B.Sc. program is only a start of a lifelong learning experience. I appreciate his consistent support from the first day I applied to the undergraduate program to these concluding moments. I am truly grateful for his progressive vision about my training in science, his tolerance of my native mistakes, and his commitment to my future carrier.

My sincere thanks go to all my lab mates, members of the staff, and the Information and Communication Engineering Department, NSTU, who helped me in many ways and made my stay at NSTU pleasant and unforgettable.

I acknowledge my sincere indebtedness and gratitude to my parents for their love, dream, and sacrifice throughout my life. Special thanks should be given to my committee members. I would like to acknowledge their comments and suggestions which were crucial for the successful completion of this study.

ABSTRACT

Computer-based evaluation of answer scripts is becoming increasingly important in the field of education. In this thesis, we propose a computer-based answer script evaluation system that evaluates both text and figures in answer scripts. For text evaluation, we consider two cases and use different methods for each case, including writing score, keyword matching score, Cosine similarity score, NLP-based cosine similarity score, and plagiarism matching score. Our system achieves high accuracy scores of 83% and 85% for each case, respectively. For figure evaluation, we use cosine similarity and Euclidean distance techniques, achieving accuracy scores of 85% and 72%, respectively. Our system provides a robust and reliable way to evaluate answer scripts in an automated manner. By using a combination of different evaluation techniques, we are able to effectively evaluate both text and figures, improving the accuracy of the overall evaluation. This system has the potential to greatly reduce the workload of educators, while also providing students with more immediate feedback on their work. Overall, our system represents a significant step forward in the field of automated answer script evaluation.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

AI	Artificial Intelligence
ML	Machine Learning
CNN	Convolution Neural Network
NLP	Natural Language Processing
RAM	Random Access Memory
NLTK	Natural Language Toolkit
LSA	Latent Semantic Analysis
SVD	Singular Value Decomposition
JPEG	Joint Photographic Expert Group
CSV	Comma-Separated Values
LSI	Latent Semantic Indexing
TF-IDF	Term Frequency – Inverse Document Frequency
WMD	Word Mover Distance
KNN	K-Nearest Neighbors
OWA	Ordered weighted Average
FCA	Formal Concept Analysis
CWE	Common Written Examination
RNN	Recurrent Neural Network
CRNN	Convolution Recurrent Neural Network
LSTN	Long – Short Term Memory
NN	Neural Network

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The process of evaluating students' performance has always been an essential aspect of the education system. The traditional approach of evaluating students' academic progress is based on manual grading of their assignments and exams, which can be time-consuming and prone to human errors. However, recent advancements in artificial intelligence (AI) and machine learning (ML) techniques have led to the development of more efficient and accurate methods for evaluating students' performance.

In this thesis, we focus on the development of automated techniques for evaluating students' performance based on the correct use of grammar, and textual content, as well as their ability to create graphs that match the expected ones. Specifically, we explore four different techniques for calculating text similarity: Common keyword matching, Cosine similarity, Plagiarism check, and NLP-based Cosine similarity. We also utilize the Grammarly tool to evaluate the grammatical correctness of students' written content.

In addition, we investigate the use of convolutional neural networks (CNN) for handwriting detection. Finally, we explore a technique for comparing student-created graphs with expected ones. The technique is Cosine similarity.

The goal of this thesis is to develop computer-based evaluation techniques that are more accurate, efficient, and reliable than traditional manual grading methods. By automating the evaluation process, we aim to reduce the workload on teachers, enable faster feedback for students, and improve the overall quality of education.

Overall, this thesis contributes to the growing body of research on the use of AI and ML in education and offers valuable insights into the development of automated evaluation techniques for students' performance.

1.2 BACKGROUND STUDY

Assessing student performance and providing feedback is an integral part of the educational process. However, traditional methods of assessment, such as written exams and assignments, can be time-consuming and subjective, leading to inconsistencies in grading. With the advancement of technology, there has been a growing interest in using computer-based methods for student assessment.

In recent years, there has been a significant amount of research focused on developing and improving automatic assessment systems. These systems use various techniques, such as natural language processing, machine learning, and computer vision, to evaluate student performance.

One area of interest in student assessment is handwriting recognition. With the rise of digital devices, the ability to recognize handwriting can be used to assess student performance in various tasks, such as essays, math problems, and language assignments.

Another area of interest is in evaluating the grammar of the student-written text. While grammar checkers like Grammarly have been widely used, they often have limitations and may not catch all errors in a student's work.

In addition to handwriting recognition and grammar evaluation, there has also been researching on comparing student answers to expected answers. This involves techniques such as common keyword matching, cosine similarity, NLP-based cosine similarity, and plagiarism check. By comparing student answers to expected answers, teachers can quickly identify areas where students may need additional instruction or support.

Finally, assessing student performance in graphing tasks has also gained interest. This involves comparing the student's graph to an expected graph using techniques such as cosine similarity, Euclidean distance, and pixel analysis. This can be particularly useful in evaluating student performance in math and science courses.

1.3 PROBLEM STATEMENT

Despite the numerous benefits of computer-based assessment systems, there are still challenges that need to be addressed. The purpose of this thesis is to explore various techniques for assessing student performance and to determine the effectiveness of these techniques in providing accurate and consistent feedback.

The issue is the lack of precise and effective methods for evaluating student performance across a range of academic disciplines. Conventional evaluation techniques, such multiple-choice exams and essay writing, aren't always effective at evaluating specific abilities, like handwriting, graphing, and textual similarity. These restrictions may result in flawed assessments, which could have detrimental effects on both students and teachers. The evaluation may occasionally be impacted by the emotion and mood of the educator.

Specifically, this thesis will focus on handwriting detection using a CNN model, grammar evaluation using Grammarly, and comparison of student answers to expected answers using common keyword matching, cosine similarity, NLP-based cosine similarity, and plagiarism check. Additionally, this thesis will explore the effectiveness of using cosine similarity and Euclidean distance for assessing student performance in graphing tasks.

By addressing the limitations of current assessment methods and exploring new techniques, this thesis aims to provide valuable insights into the development of more effective and accurate automatic assessment systems for educators.

1.4 MOTIVATION

The motivation for the problem lies in the need for accurate and efficient methods to assess student performance in various educational domains. Traditional methods of evaluation, such as multiple-choice tests and essay writing, have limitations in assessing certain skills such as handwriting, graphing, and textual similarity. These limitations can lead to inaccurate evaluations, which can have negative consequences for both students and educators. Sometimes educator’s emotion and state of mood may affect the evaluation.

Detecting and assessing student work is an essential aspect of education, particularly in today's digital age. With the widespread availability of technology, students are increasingly submitting their work electronically, resulting in a massive amount of data that educators must process. Grading assignments manually can be time-consuming and can lead to inconsistent results due to human error. Hence, developing an automated system to evaluate student work can greatly reduce the time and effort required by educators while also improving the accuracy and fairness of grading.

In this context, handwriting detection, text similarity calculation, and graph matching are critical tasks that can help automate the evaluation process. Handwriting detection can be used to verify the authorship of a document or to distinguish between handwritten and typed text. Text similarity calculation can determine how closely a student's answer matches the expected answer, and graph matching can determine how closely a student's graph matches the expected graph. These techniques can help educators quickly identify areas where students need improvement and provide personalized feedback to help them improve their performance.

Furthermore, there is an increasing need to combat academic dishonesty, particularly in the form of plagiarism. With the widespread availability of online resources, it has become easier for students to plagiarize and submit work that is not their own. Therefore, the use of plagiarism detection techniques can help educators identify instances of academic dishonesty and take appropriate action.

The development of automated systems for evaluating student work can greatly benefit the education sector by increasing efficiency and fairness while also promoting academic integrity.

1.5 OBJECTIVES

The specific objectives of this research include:

1. To make everyone believe that descriptive answer sheets can be evaluated with machines.
2. To develop an automated system that can accurately assess student responses and provide immediate feedback to instructors, saving time and resources.
3. To evaluate the effectiveness of these techniques in terms of accuracy, efficiency, and scalability.

4. To demonstrate the practical applications of these techniques in fields such as education, document analysis, and automated grading systems.
5. To provide educators with an efficient and accurate tool for grading student work, which can save time and reduce subjectivity.

1.6 RESEARCH SCOPE

Different types of tools will be used in this research work. Some are listed below.

- **Hardware requirement:** A Desktop or a Laptop with the minimum feature of core i5, 8GB RAM, 4GB graphics card
- **Software requirement:**
    1. Jupyter Notebook
    2. Google Colab
    3. Python 3.10.7
    4. MS office 2019
    5. Google Docs

1.7 ORGANIZATION OF THE THESIS

This thesis consists of 5 chapters and these chapters are organized as follows:

Chapter 1 - Introduction: Performs the background of the Answer evaluation system. In addition, the problem statement, motivation for the problem, objectives, and the scope of the study is presented.

Chapter 2 - Literature Review: An illustrative review of related work is performed on the Handwritten and typed answer evaluation system.

Chapter 3 - Methodology: Presents multiple approaches to implement a model of the answer evaluation system.

Chapter 4 - Results Analysis: Discussed the result and make comparisons among various techniques to find the best one.

Chapter 5 - Conclusions: This chapter describes the conclusion and future research scope.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

The literature review is an essential component of any research study, providing a comprehensive overview of the existing knowledge, theories, and practices related to the research topic. In this section, we will present a review of the relevant literature on the topic of our thesis, which is focused on the development of a model for handwriting detection and text analysis in educational systems. The review will cover the key studies and research works conducted by scholars in this field, highlighting their contributions and limitations, and identifying the research gaps that our study aims to address. This literature review will serve as the foundation for our research, providing us with a better understanding of the current state of the field and the potential avenues for future research.

2.1 COMPUTER COMPOSED ANSWER EVALUATION

Several works deal with computer-composed documents. In that case, the student submits their answer directly in the answer section for the corresponding Question. K Arlitsc collected the data set in the very first step which consists of answers to the questions in the question paper [17]. Upon collecting the data, all the text in the data is converted to lowercase. After the conversion to lowercase, word tokenization is performed on the text. Word tokenization is the process of splitting a large sample of text into words. This is a requirement in natural language processing tasks where each word needs to be captured and subjected to further analysis like classifying and counting them for a particular sentiment etc. The Natural Language Tool kit (NLTK) is a library used to achieve this. Moving forward, the next important step that is

performed is the removal of stop words and punctuation. A stop word is a commonly used word (such as “the”, “a”, “an”, or “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query. We would not want these words to take up space in our database, or take up valuable processing time. Shai Shalev-Shwartz, Batuhan Balci, and Judy McKimm approach almost the same process [18, 19, and 20].

Some other author works with only one sentence answer evaluation. Judy McKimm thought about only some 1line answers for some online quizzes [20]. He used the processed data is then converted into vectors using Embedding and is then passed through a cosine formula which provides the similarity between the two. Based on the value of similarity achieved (between 0 and 1), marks are rewarded for each provided answer and the final result is calculated by adding them all. G. Jain and Effie Lai-Chong Law also thought the same thing [10 and 22].

R. S. Wagh, Yuan, Zhenming, and Sophal Chao deal with some typed documents of laws to check their similarity and check plagiarism [7, 21, and 24].

2.2 HANDWRITTEN ANSWER EVALUATION

Handwritten answer evaluation is a relatively new development in the field of education and assessment, and it has the potential to revolutionize the way we evaluate student responses. Automated systems for handwritten answer evaluation use computer vision and machine learning algorithms to analyze and score handwritten responses, which can be faster, more objective, and more consistent than manual evaluation. Despite these advantages, there are still limitations to automated systems, particularly in capturing nuances in writing style and tone or dealing with messy or illegible handwriting. As such, more research is needed to explore the effectiveness of these systems in different contexts and to understand their potential impact on education and assessment practices.

2.2.1 Semantic and Synonym-Based Evaluation

R. S. Wagh (2020) and H. Mangassarian present an automated system that addresses the following problems with assessing subjective questions: synonymy, polysemy, and trickiness [7 and 30]. Latent semantic analysis (LSA) and the ontology of a subject are

introduced to solve the problems of synonymy and polysemy. A reference unit vector is introduced to reduce the problem of trickiness. The system consists of two databases: a science knowledge library and a question- and reference-answer library. The science knowledge library stores the ontology of a subject as text documents. The question- and reference-answer library stores questions as text documents and reference answers as a text document matrix. When a teacher adds new questions, a system using this science knowledge library will search for related points of knowledge and keywords and give them to the teacher. Then, the teacher will submit the reference answer to the system. It will process the reference answer using Chinese automatic segmentation, which produces text document vectors and sends them to the teacher. Then, the teacher detects the terms and their weights for each vector and sends them back to the system. Weights of the terms in the reference answer are computed using the term-frequency and inverse-document-frequency functions. In the questions and reference answers, the library will save the vector of the reference answers and questions as text documents. To compute the similarity between a student's answer and the reference answer, the former is sent to the system, which assesses the answer using Chinese automatic segmentation and produces a text vector projected into k-dimensional LSA space. This LSA is formed by a vector using the mathematical technique of singular value decomposition (SVD), which represents terms and documents that are correlated with each other. The system computes the cosine similarity of student and reference answer vectors projected into k-dimensional LSA space in the reference unit vector. Similar types of processes are followed by M. Kusner, E. Kim, Orkphol, and L. A. Cutrone. But used different algorithms and methods to get better outcomes [2,3,5, and 28].

For the online evaluation of subjective questions, Hu and Xia advocated the use of latent semantic indexing [1]. To create a k-dimensional LSI space matrix, they combined subjective ontologies with Chinese automatic segmentation algorithms. The answers were given as TF-IDF embedding matrices, and the term-document matrix was then subjected to Singular Value Decomposition (SVD), resulting in a semantic space of vectors. Synonym and polysemy issues were lessened thanks to LSI. Finally, cosine similarity was used to calculate the degree of similarity between the replies. 35

classes and 850 cases were included in the dataset, and the results revealed a 5% discrepancy between teacher grading and the suggested approach.

Due to its success in grading brief descriptive answers using the morphologically complicated Korean language, J. E. Kim devised a method (LSP) [3]. To better comprehend the user's intents, LSP can arrange the semantics of the response. The keywords were also expanded to include synonyms to make them more adaptable to different answer types. The 88-student dataset was converted to LSP and then compared to the solution LSP to get the answer's score. As a consequence, the new system outperformed the old one by 0.137.

2.2.2 Keyword Matching-Based Evaluation

That answer sheet is made available to the system by Hanumant R. Gite in jpeg (.jpg) format [25]. Provide the answer's minimum length, maximum points, and keywords. Words from the provided answer will be broken apart by the system. An A.csv file will be used to store the specified terms. Words from the CSV file will be counted to determine the length of the response. Verify the percentage of matched terms. Compare the number of words written to the required minimum. Examine the graph for the proportion of marks provided for the specified percentage of keywords that were matched. See the graph to see the proportion of marks provided for the specified % of word length. Multiply both percentages of the answer's possible score. Show the grades earned.

Using text summarizing, text semantics, and keyword summarization, Bahel and Thomas developed an architecture for the evaluation of subjective questions and contrasted the outcomes with current methods [11]. According to the data, the error was 1.372 instead of the 1.312 predicted by Jaccard's similarity technique. Nevertheless, the method did not calculate non-textual data, including diagrams, pictures, and other formats.

The author's analysis is the most accurate I've ever studied. Similar steps are taken by H. Mittal as well; however, they employ different algorithms [27].

2.2.3 KNN Classification-Based Evaluation

Word Mover's Distance (WMD) is an innovative tool that M. Kusner introduced for determining the differences between two texts [2]. The vector space constraints were

loosened by the system's adoption of a lax WMD methodology and the absence of hyper-parameters. Eight real-world datasets were added, including sentiment data from Twitter and sports stories from the BBC. Two more bespoke models were trained in addition to the Word2vec model from Google News. The testing data were classified using the KNN classification method. Hence, loosened WMD resulted in 2 to 5 times faster classification and decreased error rates.

To determine how closely two texts resemble one another based on the keywords that exist in at least one of the documents, Oghbaie and Zanjireh suggested a pair-wise Similarity measure [4]. The study offered a modified version of the preferred properties technique termed PDSM (pair-wise document similarity measure), which is a new similarity metric. The proposed similarity metric was used in text mining tasks like document recognition, K-means clustering, and k Nearest Neighbors (KNN) for single-label classification. The accuracy of the procedure was evaluated, and the PDSM method outperformed other measures like the Jaccard coefficient by 0.08 recall.

2.2.4 Cosine Similarity-Based Evaluation

To represent words on a fixed-sized vector space model using the word2vec method, Orkphol and Yang [5] employed a cosine similarity measure to assess the similarity of sentences. The sentence vector was created by averaging the words in the sentence using the Google tool Word2vec. The score was considered valid if it exceeded a predetermined similarity result threshold, which ranged from 0 to 1. The system's performance was measured by recall and accuracy and was 50.9% with and 48.7% without the probability of sense distribution.

To find commonalities between various legal papers, C. Xia coupled the word2vec method with the corpus of legal texts [6]. The similarity between several text vectors was calculated using the cosine similarity formula. As a consequence, word2vec enhanced accuracy by 0.2 when compared to the Bag of Words method. By training the word2vec model on legal documents, accuracy might be raised by another 0.05-0.10 points.

2.2.5 Concept-Based Similarity Evaluation

Wagh and Anand proposed a multi-criteria decision-making perspective to find the Similarity between legal documents [7]. The work included using Artificial

Intelligence and aggregation techniques such as ordered weighted average (OWA) for obtaining the similarity value between different documents. The dataset was obtained from Indian Supreme Court case judgments from years ranging from 1950 to 1993. Evaluation measures of score and recall were used. As a result, a concept-based similarity approach such as the one proposed in the work performed better than other techniques such as TF-IDF, getting an F1 score of up to 0.8.

To determine the context of phrases, Alian and Awajan used multiple-word embedding models, clustering algorithms, and weighting methods to study various parameters affecting sentence similarity and paraphrasing detection [8]. AraVec and FastTex, two pre-trained embeddings, were both trained for the Arabic language. About 77,600,000 tweets total were included in the Arabic training dataset. Because of this, pre-trained embedding using expert-labeled data produced improved recall and precision for K-means and agglomerative clustering, respectively, of 0.87 and 0.782.

2.3 SUMMARY OF THE RELATED WORK

A literature review is a critical analysis of the existing research on a particular topic or question. It involves identifying, analyzing, and synthesizing relevant sources to provide a comprehensive understanding of the state of knowledge on the topic. The purpose of a literature review is to evaluate and compare different perspectives, methodologies, and findings in the field, and to identify gaps, inconsistencies, and controversies that require further investigation. A well-conducted literature review is an essential component of scholarly research, as it helps researchers to contextualize their own work, identify research questions, and build on the existing knowledge in their field.

Table 2.1: The summary of the related work.

Ref.	Method used	Contribution	Limitation	Difference to us
[1]	Find the Similarity between answers was	The dataset consisted of 35 classes and 850 instances marked by teachers, and the results	Doesn't support handwritten answer sheets.	Our proposed system will work with a

	calculated using cosine similarity	showed a 5% difference in grading done by teachers and the system.	Nothing about figure evaluation	handwritten answer. We are considering figure evaluation
[2]	Using Word Mover's Distance (WMD)	Find the dissimilarity between the two texts. WMD reduced the error rates and led to 2 to 5 times faster classification.	No Evaluation, just measures of dissimilarity	Evaluates considering the similarity between the correct answer and the student's answer
[4]	PDSM (pair-wise document similarity measure), k Nearest Neighbors (kNN) for single-label classification.	proposed a pair-wise Similarity measure to measure the similarity between two documents based on the keywords which appear in at least one of the documents.	Nothing about figure evaluation	We are considering figure evaluation
[6]	combined the word2vec approach with the legal document corpus. Cosine similarity was used to measure the Similarity between	identify similarities between different law documents.	Doesn't support handwritten answer sheets. Nothing about mathematical expression and figure evaluation	Our proposed system will work with a handwritten answer. We are figure evaluation. Our System will work in the education

	different sentence vectors			sector, not in the law sector
[7]	Evaluation measures of F1 score and recall were used	The work included using Artificial Intelligence and aggregation techniques such as ordered weighted average (OWA) for obtaining the similarity value between different documents.	Doesn't support handwritten answer sheets. Nothing about mathematical expression and figure evaluation	Our proposed system will work with a handwritten answer. We are considering figure evaluation. Our System will work in the education sector, not in the law sector
[8]	using a different word embedding models, clustering algorithms, and weighting methods to find the context of sentences	studied various factors affecting sentence similarity and paraphrasing identification	Doesn't support handwritten answer sheets. Nothing about mathematical expression and figure evaluation	Our proposed system will work with a handwritten answer. We are considering figure evaluation. Our System will work in the education sector.
[9]	using formal concept analysis (FCA)	The proposed system detected plagiarism in documents with 94% accuracy.	Doesn't support handwritten answer sheets. Limited scope.	Our System will work in the education sector for Subjective

				answer evaluation.
[10]	A novel approach for subjective questions evaluation using concept graphs.	The score was evaluated using various graph similarity techniques.	Nothing about mathematical expression. Similar words may not be detected. Word sentiment not in concern	Proper statements of Similar words may be detected. Word sentiment is a concern
[13]	translate handwritten text to a format that is machine readable	This paper is using the artificial neural network to achieve high accuracy for optically recognizing the character.	Nothing about mathematical expression and figure evaluation	We are considering figure evaluation

CHAPTER – 3

METHODOLOGY

3.1 INTRODUCTION

The methodology section of this thesis outlines the step-by-step process used to conduct the research and analyze the data. It describes the approach taken to address the research questions and achieve the research objectives. The methodology includes a detailed explanation of the overall architecture of the system, data collection process, data pre-processing, the tools and techniques used for data analysis. The aim of the methodology is to provide a transparent and rigorous account of the research process to ensure that the results obtained are valid, reliable, and generalizable to the population of interest. The following section presents the methodology used in this study.

3.2 ARCHITECTURE AND FLOW DIAGRAM

The diagram may include boxes or nodes representing each step in the process, with arrows indicating the flow of work between them. It can also be useful to include annotations or descriptions of each step to provide more detail or clarify any potential confusion.

3.2.1 Architecture

This architecture shows how each component is connected to the other. The Exam section will provide question papers to the students and students submit their responses to the exam section.

Then the answer sheets provided by the students need to scan and convert each page of the answer sheet to an image (format: .jpg). Those compressed images will be the input of our proposed model. This Answer Evaluation System will generate a score after evaluating the paper by performing some specific techniques which will be properly discussed in this chapter. After that, the score will be stored in a database controlled by the Exam Section. Student can collect their result from that database after publishing the result.

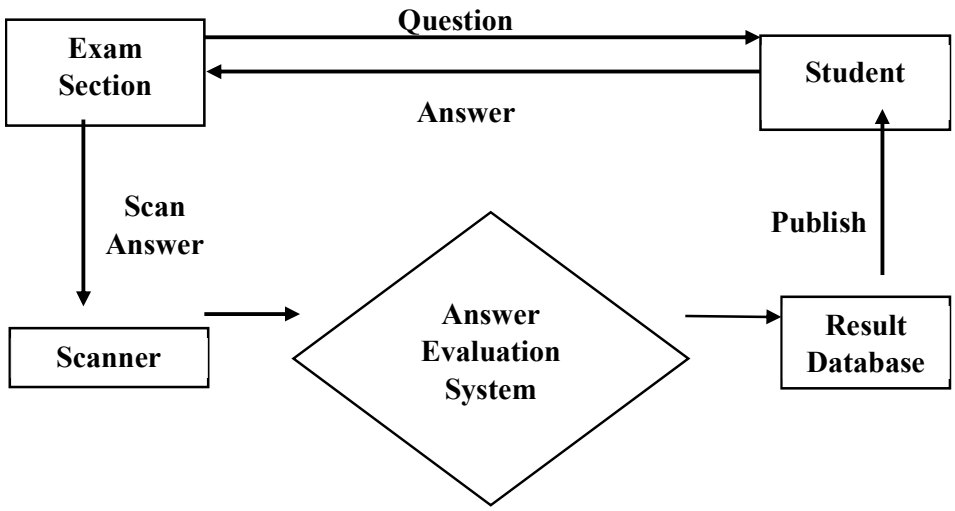
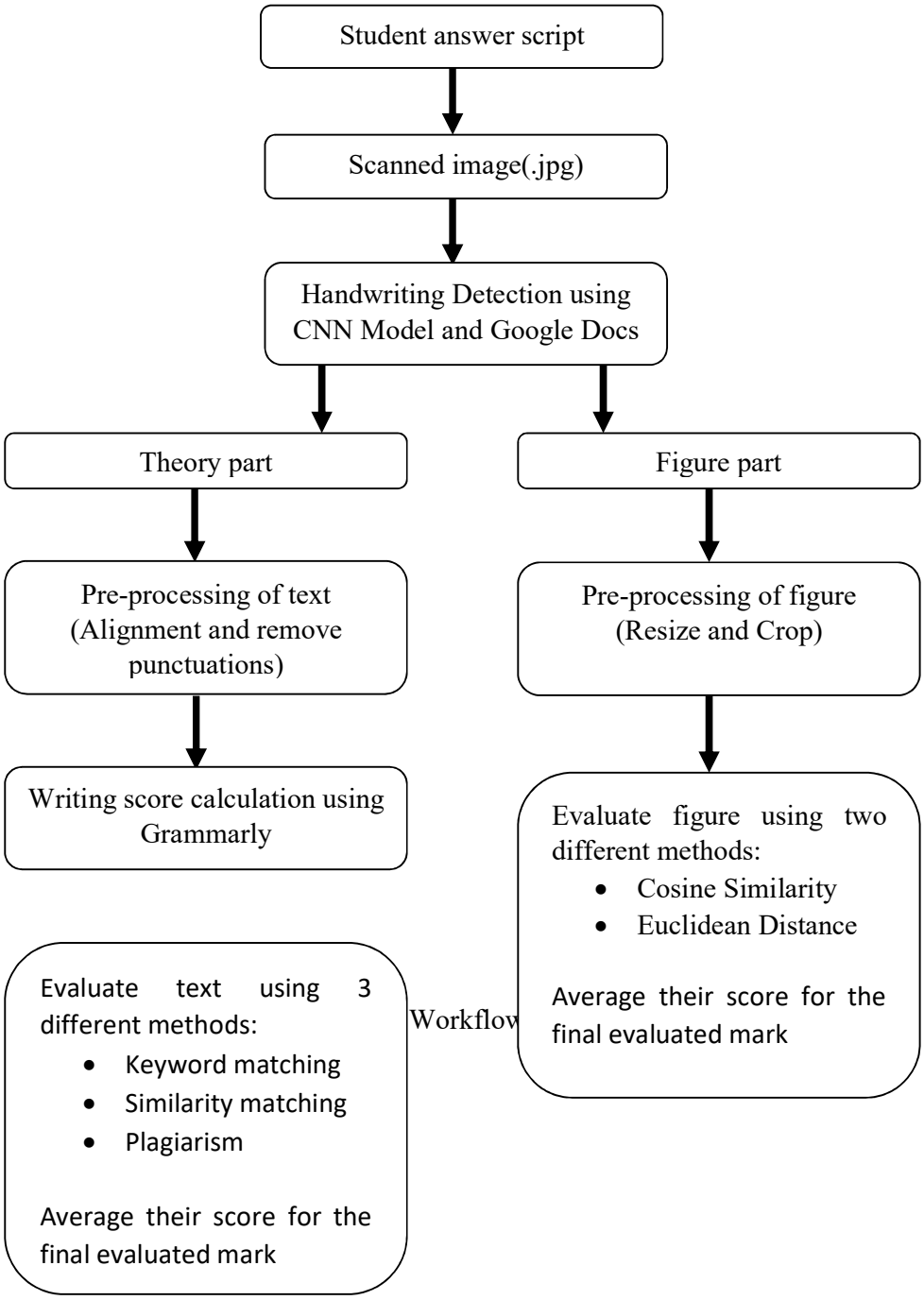


Figure 3.1: Overall architecture of the system

3.2.2 Workflow

A workflow diagram is a visual representation of the steps and tasks involved in a process. It can be used to illustrate the flow of work, identify potential bottlenecks or inefficiencies, and ensure that all necessary steps are included. In the context of your thesis, a workflow diagram can be used to illustrate the steps involved in the answer script evaluation process using the techniques you have described, such as handwriting detection and text and figure preprocessing

A well-designed workflow diagram can help to ensure that the evaluation process is clear, efficient, and reproducible.



Workflow

3.3 ANSWER SCRIPT FORMAT

In the context of student assessment, it is essential to maintain consistency and standardization in the answer scripts. Therefore, the formatting and pre-arrangement of the answer scripts are crucial aspects to consider in any automated evaluation

system. In this section, we discuss the formatting requirements for the student answer scripts that were considered in our thesis. The answer script should be written in a specific format to ensure that it is easy to parse and analyze. The format includes three different types of answer scripts:

The answer script should be written in a specific format to ensure that it is easy to parse and analyze. The format includes three different types of answer scripts:

- **Only text:** The answer must start from e new page.
- **Only Figure:** must be on a new page
- **Text + figure:** The answer must start from a new page as well as the figure also start on a new page.

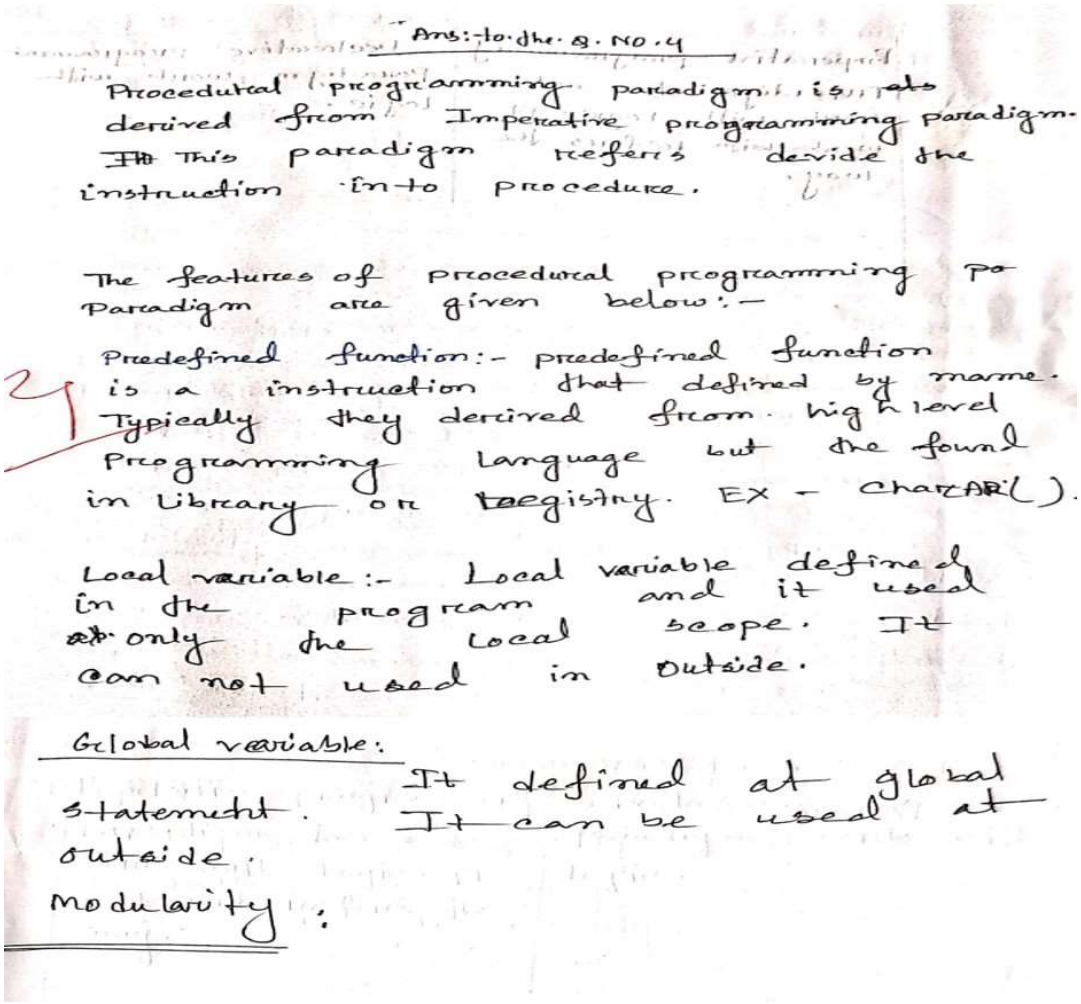
3.4 DATA COLLECTION

The data for this study was collected from the answer sheets of students who had appeared for an exam in a particular subject. A total of around 100 answer scripts were collected from students belonging to different classes and academic backgrounds. These answer sheets were then scanned and digitized to create a digital dataset for further analysis. The dataset included the handwritten responses of the students as well as the expected answers provided by the examiners. In addition to this, demographic information such as the student's name, roll number, and the class was also recorded. The data collection process was carried out in a controlled environment to ensure that the answer sheets were not tampered with and the data was accurate and reliable. The collected data formed the basis of the study and was used to train and test the proposed model.

For collecting those data there is permission from relevant authorities.

Sample of answer scripts:

Sample 1



Ans:- to the Q. NO. 4

Procedural programming paradigm is derived from Imperative programming paradigm. This paradigm refers devide the instruction into procedure.

The features of procedural programming Paradigm are given below:-

Predefined function:- predefined function is a instruction that defined by name. Typically they derived from high level Programming language but the found in library or registry. EX - charAR().

Local variable:- Local variable defined in the program and it used only the local scope. It can not used in outside.

Global variable:- It defined at global statement. It can be useed at outside.

Modularity:

Sample 2

Ans. to the Q. NO- 4

The features of Procedural programming paradigm are.

Predefined Functions:

Predefined Functions is typically an instruction identified by name. It is built in high level programming language but they are derived from the library and the string. An example of predefined function is 'charAt'.

Local variable:

Local variable is those variable which are declared in any main structure of any method and local scope is given in it.

Global variable:

Global variable is those which is declared outside any function in any code. Global variable works for the whole code unlike, the local variable.

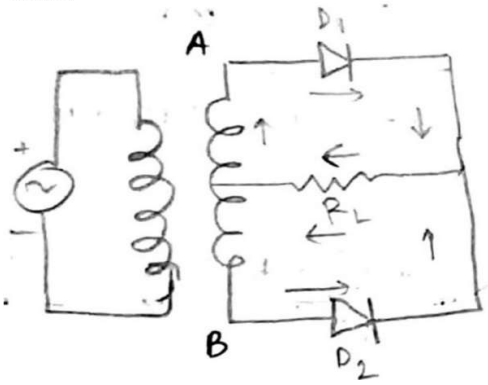
modularity:

modularity is a system where every system have different works at hand but are grouped together to solve any larger problem.

Parameter passing:

Parameters passing is

Sample 3



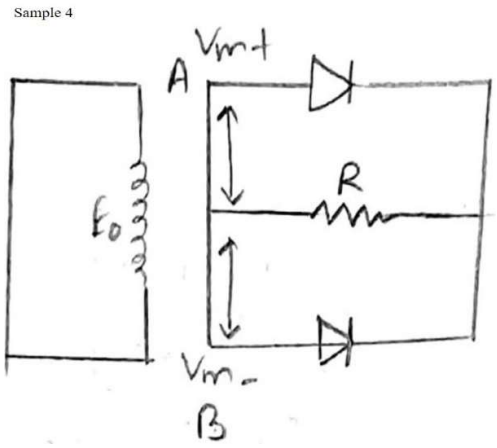


Figure 3.3: Sample answer scripts

3.5 HANDWRITING DETECTION

This task can be challenging due to variations in handwriting styles, different ink colors and pen types, and the presence of noise and artifacts in the image. However, using advanced machine learning techniques like convolutional neural networks (CNNs), it is possible to accurately detect and extract text from handwriting images. Another available way to detect handwriting is image-to-text conversion using Google docs.

3.5.1 Convolutional Neural Networks for Handwriting Detection

The use of Convolutional Neural Networks (CNN) for handwriting detection has shown promising results in recent years. CNN are a type of deep-learning neural network that can effectively capture spatial information in images. These networks are trained on a large dataset of handwriting images and learn to automatically identify and extract features that are relevant for handwriting detection. By using a CNN model, your thesis can achieve high accuracy in detecting and extracting text from handwritten images, which can help in automating the process of evaluating student answer scripts. Here is a step-by-step process for handwriting detection using a Convolutional Neural Network (CNN):

Data Collection: Collect a dataset of handwritten characters or words. This dataset should include both training and testing data. This dataset is collected from students' answer scripts.

Data Preprocessing: Preprocess the dataset to ensure that it is in a format suitable for the CNN model. This includes resizing the images, converting them to grayscale, and normalizing the pixel values.

Model Architecture: Choose an appropriate CNN architecture for your handwriting detection task. This can include a combination of convolutional layers, pooling layers, and fully connected layers.

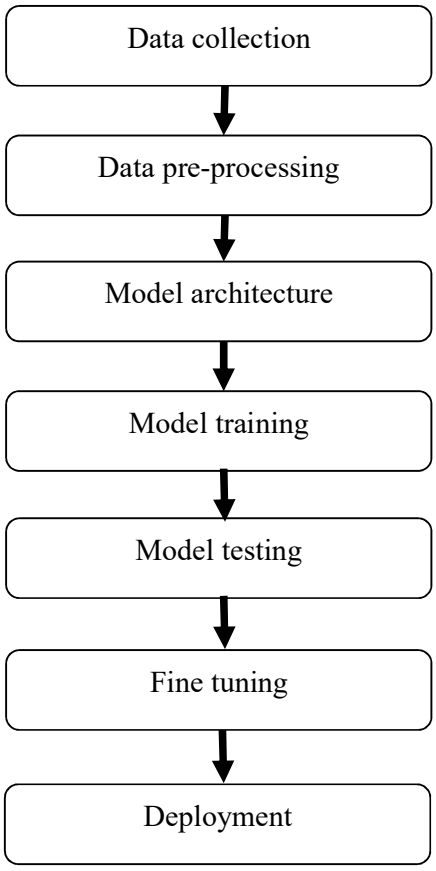


Figure 3.4: Workflow of Convolutional Neural Network model

Model Training: Train the CNN model using the preprocessed training dataset. Use techniques like data augmentation and dropout to improve the generalization ability of the model. Here 80% data is used for training.

Model Testing: Test the trained CNN model using the preprocessed testing dataset. Evaluate the performance of the model using metrics like accuracy, precision, recall, and F1 score. The remaining 20% data is used for testing.

Fine-tuning: Fine-tune the CNN model by adjusting hyperparameters like learning rate, batch size, and the number of epochs. Also, experiment with different CNN architectures to improve the performance of the model.

Deployment: Deploy the trained CNN model for handwriting detection in a real-world application. This can include developing a web application or mobile app that allows users to upload an image of handwritten text and receive the recognized text as output. Overall, this process involves data collection, preprocessing, model architecture selection, training, testing, fine-tuning, and deployment of a CNN model for handwriting detection

3.5.2 Handwriting Detection Using Google Docs

Handwriting detection using Google Docs is a convenient and accessible way to convert handwritten notes or documents into digital text. Google Docs uses Optical Character Recognition (OCR) technology to recognize and convert handwriting into machine-readable text. This technology enables users to quickly and easily create digital copies of handwritten notes or documents that can be easily edited, shared, and searched.

To use handwriting detection in Google Docs, users can simply open a new or existing document and select the "Tools" option from the menu bar. From there, they can select "Enhanced Scans" and then choose "Handwriting" to activate the handwriting detection feature. Once activated, users can use their device's touchscreen or a stylus to write or draw directly onto the document, and Google Docs will automatically convert the handwritten text or drawings into digital text.

Handwriting detection using Google Docs has several advantages, including its ease of use, accessibility, and compatibility with a wide range of devices. However, it is important to note that the accuracy of handwriting detection can vary depending on the quality of the handwriting, the device used, and other factors. Additionally, handwriting detection may not be suitable for all types of documents or notes, and users should consider the specific needs of their project before deciding whether to use this technology.

3.6 ANSWER SCRIPT EVALUATION

The evaluation of the answer script using the proposed techniques involves a comparison of the extracted text and figures from the student answer sheet with the expected answers. The comparison is done by using various techniques such as matching common keywords, cosine similarity, plagiarism check, and NLP-based cosine similarity. The technique that provides the highest similarity score is used to evaluate the answer script.

Additionally, the graph matching technique is used to compare the student's graph with the expected graph. This comparison is done by using cosine similarity and Euclidean distance. The technique that provides the highest similarity score is used to evaluate the answer script.

3.6.1 Text Evaluation

Text evaluation is a critical part of the answer script evaluation process. In our technique, we use several methods to evaluate the text extracted from the student's answer script. First, we remove all punctuation marks and unwanted symbols from the text, ensuring that only the relevant content is evaluated. Then, we convert all text to black color to ensure consistency in the evaluation process.

Next, we use several techniques to evaluate the text content. These include the following techniques.

- Keyword matching score
- Similarity matching score, and
    - cosine similarity, and
    - NLP-based cosine similarity.
- Plagiarism check score

The common keyword method compares the extracted text with a list of possible keywords and their synonyms to identify relevant content. Cosine similarity and NLP-based cosine similarity evaluate the similarity between the extracted text and the expected answer, while the plagiarism check detects any instances of copied content.

Here text evaluation sectorize into two cases:

- *First case:* Averaging writing score, keyword matching score, similarity score by cosine similarity, and plagiarism check score. Generation of final mark in this case will be done by averaging the mark of these 4 techniques.
- *Second case:* Averaging writing score, keyword matching score, similarity score by NLP-based cosine similarity, and plagiarism check score. Generation of final mark in this case will be done by averaging the mark of these 4 techniques.

3.6.1.1 Keyword matching score

Keyword matching means finding the percentage of expected keyword present in the students' answer scripts. This expected keyword should be provided by the examiner or related authority.

This method includes the following steps:

- Store the text of the student's answer sheets as a text file (.txt).
- Store the list of expected keywords provided by the teacher as a text file (.txt).
- Calculate writing score from Grammarly maintaining table 3.1.
- Remove all punctuation marks and unwanted symbol from both text file.
- Read both the text file as a set of strings where each word acts as an element of the set.
- Calculate the percentage of the expected keyword found in the student answer sheet.
- Generate a score (out of 5) according to the percentage by maintaining table 3.2.

Python code for generate keyword matching score:

This code shows all words present in the text files get separate and stored in a vector of string. In stripped1 all words of students answer scripts were stored, and in stripped2 all words of expected answer were stored. Then finding the percentage of expected keyword present in the students' answer scripts.

```
import string
with open ("001.txt") as file1, open ("expected keyword.txt") as file2:
    s1=file1.read()
    s2=file2.read()
```

```
words1=s1.split()
table1= str.maketrans("", "", string.punctuation)
stripped1 = [w.translate(table1)for w in words1]
l1=(len(stripped1))
print(stripped1)
```

```
words2=s2.split()
table2= str.maketrans("", "", string.punctuation)
stripped2 = [w.translate(table1)for w in words2]
l2=(len(stripped2))
print(l2)
print(stripped2)
```

```
ans=[]
for i in stripped1:
    if i in stripped2:
        ans.append(i)
```

```
print(ans)
percentage_matched = len(ans)/l2*100
print(percentage_matched)
```

Table 3.1: Scoring System for Correct writing

Percentage (%)	Mark (out of 5)
>=80	5
>=60	4
>=50	3
>=30	2
>=10	1

Table 3.2: Scoring system for keyword matching score

Percentage (%)	Score (out of 5)
>=70	5
>=50	4
>=40	3
>=30	2
>=10	1

Figure 3.5: Code for keyword matching score

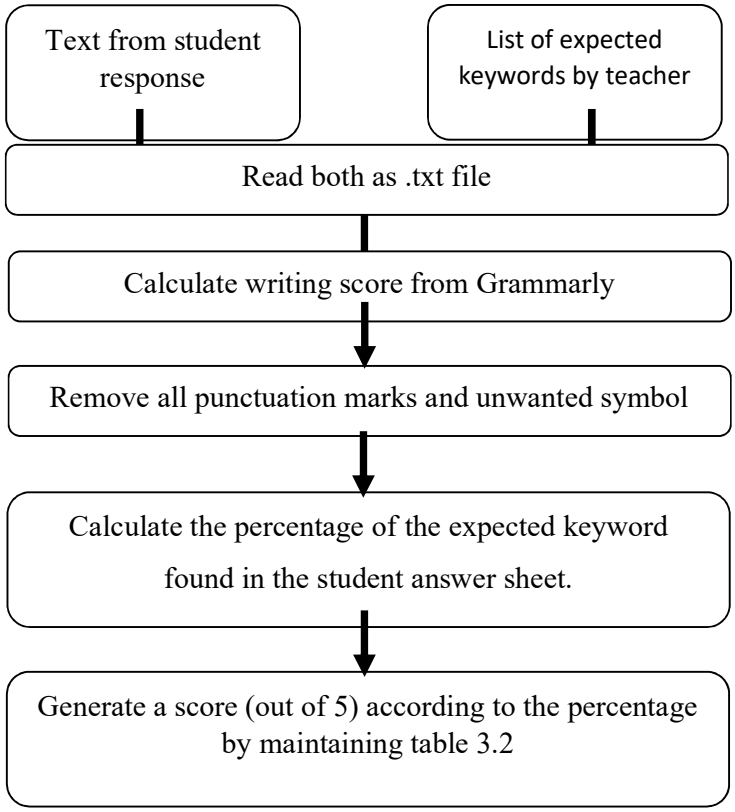


Figure 3.6: Workflow of keyword matching score

3.6.1.2 Similarity matching score

Similarity score is a crucial component of any system that aims to compare and evaluate textual data. It measures the degree of similarity between two pieces of text and is calculated using various techniques, such as cosine similarity and NLP-based similarity algorithms. A higher similarity score indicates that the two pieces of text are more similar, while a lower score indicates greater differences between them.

(i) Cosine similarity: It is a measure of the similarity between two vectors of values, often used in natural language processing and information retrieval to compare documents or text. It measures the cosine of the angle between two vectors, which indicates how similar the direction of the two vectors is.

Cosine similarity is calculated by taking the dot product of two vectors and dividing it by the product of their magnitudes. The resulting value ranges from -1 to 1, where a value of 1 indicates that the two vectors are identical in direction and a value of 0 indicates that the two vectors are orthogonal or completely dissimilar in direction.

Cosine similarity is particularly useful for comparing text documents, where each document can be represented as a vector of word frequencies or embeddings. By computing the cosine similarity between two document vectors, it is possible to determine how similar the documents are in terms of the words they contain and the topics they cover.

Cosine similarity has several advantages over other similarity measures, such as Euclidean distance or Jaccard similarity. One advantage is that it is insensitive to the length of the vectors, which means that the measure is not affected by the size of the documents or the number of words they contain. Another advantage is that it can handle sparse vectors, where most of the entries are zero, which is common in natural language processing applications.
The formula for computing cosine similarity between two vectors, A and B, is:

$$\text{Cos}(A, B) = A \cdot B / ||A|| * ||B|| \quad (3.1)$$

Or,

$$\text{cos\_sim}(A, B) = \text{dot\_product}(A, B) / (\text{mag}(A) * \text{mag}(B)) \quad (3.2)$$

where "dot_product" is the dot product of the two vectors, and "mag" is the Euclidean norm or magnitude of the vector. The dot product of two vectors can be computed as the sum of the products of their corresponding elements, while the magnitude of a vector can be computed as the square root of the sum of the squares of its elements.
This method includes the following steps:

- Store the text of the student answer sheet as a text file (.txt).
- Store the list of expected keywords provided by the teacher as a text file (.txt).
- Read both of them as a .txt file.
- Create the vectorizer for both of them. As a vectorizer here *TfidfVectorizer()* is used.

- Fit and transform the vectorizer to the text
- Calculate the cosine similarity between the texts according to the formula.
- Generate a score (out of 5) according to the similarity score by maintaining table 3.3.

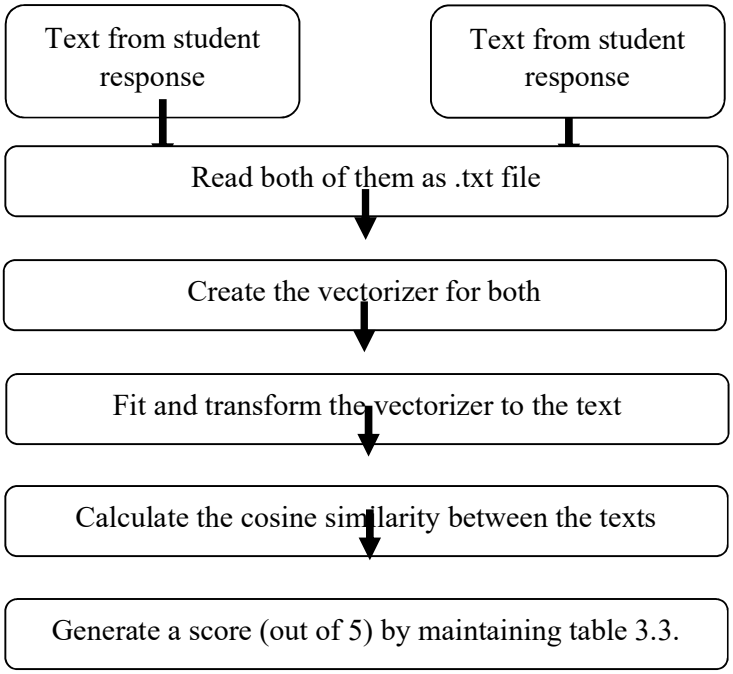


Figure 3.7: Workflow of cosine similarity for text evaluation

Table 3.3: Scoring System for Cosine Similarity technique for text

Similarity score	Score (out of 5)
>=40	5
>=25	4
>=15	3
>=10	2
>=5	1

(ii) NLP-based Cosine similarity: NLP is a field of computer science and artificial intelligence that deals with the interaction between computers and humans in natural language. It involves various tasks such as text classification, sentiment analysis, machine translation, and question-answering systems.

en_core_web_lg: en_core_web_lg is a pre-trained statistical model for the English language, developed by the spaCy natural language processing library. It is based on a neural network architecture and is trained on a large corpus of text data, including web pages, news articles, and other sources.
The "en" in "en_core_web_lg" stands for English, while "core" refers to the fact that it is a core model that includes common linguistic features such as part-of-speech tagging, dependency parsing, and named entity recognition. "Lg" stands for "large", indicating that this model includes a larger vocabulary and more complex linguistic features than smaller spaCy models.

The en_core_web_lg model can be used for various natural language processing tasks, including text classification, named entity recognition, sentiment analysis, and text summarization. It has a high accuracy rate and is considered to be one of the best pre-trained models for English language processing.

In NLP, models like en_core_web_lg are used to process and analyze natural language text data, and to extract useful information from it. These models are trained on large datasets of annotated text, and they use advanced algorithms such as deep learning to learn the patterns and structures of natural language. Once trained, these models can be used to perform various NLP tasks with high accuracy and efficiency.

This method includes the following steps:

- Store the text of the student answer sheet as a text file (.txt).
- Store the list of expected keywords provided by the teacher as a text file (.txt).
- Read both of them as a .txt file.
- Create the vectorizer for both of them. As a vectorizer here *TfidfVectorizer()* is used.
- Load “en_core_web_lg” using *spacy.load("en_core_web_lg")* function.
- Calculate the cosine similarity between the texts.
- Generate a score (out of 5) according to the similarity score by maintaining table 3.4.

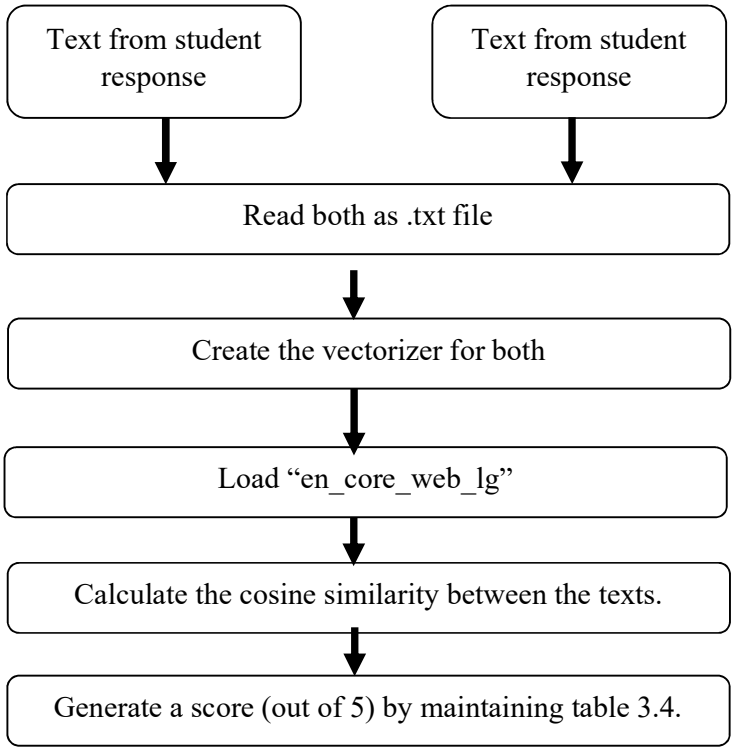


Figure 3.8: Workflow of NLP-based cosine similarity

Table 3.4: Scoring system NLP-based cosine similarity

Similarity score	Score (out of 5)
>90	5
>88	4
>86	3
>85	2
>83	1

3.7.1.3 Plagiarism matching Score

A plagiarism check refers to the process of identifying instances of plagiarism in a piece of writing or content. Plagiarism is the act of using someone else's work or ideas without proper attribution or permission, and it is considered unethical and a violation of intellectual property rights. Plagiarism checks can be done manually or with the help of plagiarism detection software tools. Manual plagiarism checks compare the

content against other sources and look for similarities or verbatim copying of text. This process can be time-consuming and may not be able to detect all instances of plagiarism.

On the other hand, plagiarism detection software tools use advanced algorithms and machine learning techniques to compare the content against a large database of sources and identify any similarities or matches. These tools can quickly identify instances of plagiarism and provide a detailed report of the findings.

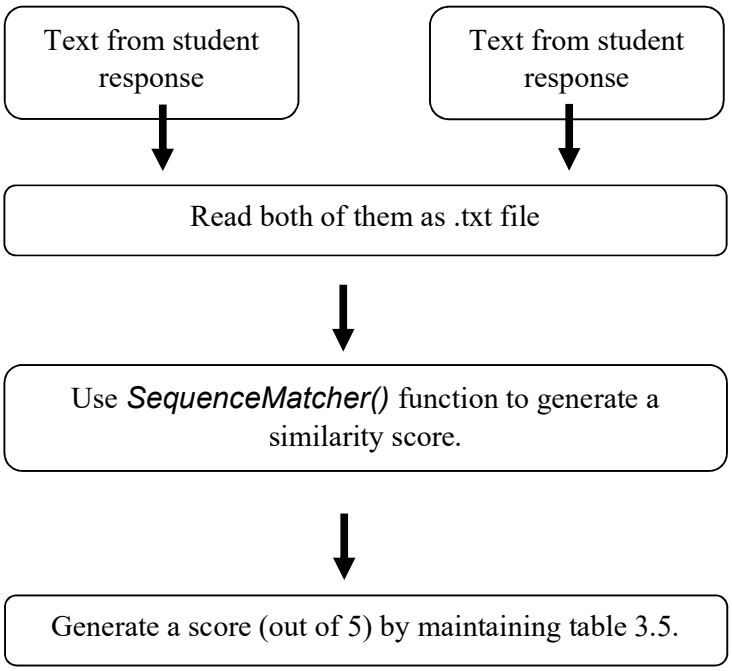


Figure 3.9: Workflow of plagiarism matching score

There are various plagiarism detection software tools available, both free and paid, that can be used to check for plagiarism. Some popular tools include Grammarly, Turnitin, Copyscape, and Plagiarism Checker. These tools can be used by students, teachers, writers, and publishers to ensure that the content is original and does not violate any copyright laws.

It is important to note that a plagiarism check is not only about avoiding legal consequences, but also about upholding academic integrity and maintaining high

standards of ethical conduct. By checking for plagiarism, individuals can ensure that their work is original and that they are giving proper credit to the sources they use. This method includes the following steps:

- Store the text of the student answer sheet as a text file (.txt).
- Store the list of expected keywords provided by the teacher as a text file (.txt).
- Read both of them as a .txt file.
- use *SequenceMatcher(None, file1_data, file2_data).ratio()* function to generate a similarity score.
- Generate a score (out of 5) according to the similarity score by maintaining table 3.5.

Table 3.5: Scoring system for plagiarism matching score

Similarity score	**Score (out of 5)**
>=6	**5**
>=5	**4**
>=4	**3**
>=3	**2**
>=2	**1**

3.6.2 Figure Evaluation

Cosine similarity and Euclidean distance are two widely used techniques for measuring the similarity between two images. Cosine similarity measures the cosine of the angle between two vectors, which can be used to compare the shapes and orientations of images. Euclidean distance measures the distance between two points in a multi-dimensional space, which can be used to compare the size and overall structure of images. In the context of figure evaluation, these techniques can be used to compare the student's graph or image with the expected graph or image. By calculating the cosine similarity or Euclidean distance between the two, we can determine how similar they are and therefore evaluate the accuracy of the student's answer. These techniques are particularly useful in situations where the expected graph or image has a specific structure or pattern that the student's answer must conform to.

Overall, the use of cosine similarity and Euclidean distance for figure evaluation can significantly improve the accuracy of answer evaluation in educational settings

3.6.2.1 Using Cosine similarity

Calculate the similarity score between the student graph and expected graph using cosine similarity. Pre-process the data to convert the graph into vector form and calculate the cosine similarity between the vectors.

Cosine similarity is a measure of the similarity between two vectors of values, often used in natural language processing and information retrieval to compare documents or text. It measures the cosine of the angle between two vectors, which indicates how similar the direction of the two vectors is

Cosine similarity is calculated by taking the dot product of two vectors and dividing it by the product of their magnitudes. The resulting value ranges from -1 to 1, where a value of 1 indicates that the two vectors are identical in direction and a value of 0 indicates that the two vectors are orthogonal or completely dissimilar in direction.

Cosine similarity is particularly useful for comparing text documents, where each document can be represented as a vector of word frequencies or embeddings. By computing the cosine similarity between two document vectors, it is possible to determine how similar the documents are in terms of the words they contain and the topics they cover.

Cosine similarity has several advantages over other similarity measures, such as Euclidean distance or Jaccard similarity. One advantage is that it is insensitive to the length of the vectors, which means that the measure is not affected by the size of the documents or the number of words they contain. Another advantage is that it can handle sparse vectors, where most of the entries are zero, which is common in natural language processing applications.

The formula for computing cosine similarity between two vectors, A and B, is:

$$\text{Cos}(A, B) = A \cdot B / ||A|| * ||B|| \quad (3.3)$$

Or,

$$cos\_sim(A, B) = dot\_product(A, B) / (mag(A) * mag(B)) \quad (3.4)$$

where “dot_product” is the dot product of the two vectors, and “mag” is the Euclidean norm or magnitude of the vector. The dot product of two vectors can be computed as the sum of the products of their corresponding elements, while the magnitude of a vector can be computed as the square root of the sum of the squares of its elements.

This method includes the following steps:

- Store the figure image from the student answer sheet as a .jpg file.
- Store the expected Figure image provided by the teacher as a .jpg file.
- Read both of them as a .jpg file.
- Define TensorVector class by which the image will be vectorized.
- Vectorize both images using an object of TensorVector
- Calculate the cosine similarity between the image according to the formula.
- Generate a score (out of 5) according to the similarity score by maintaining table 3.6.

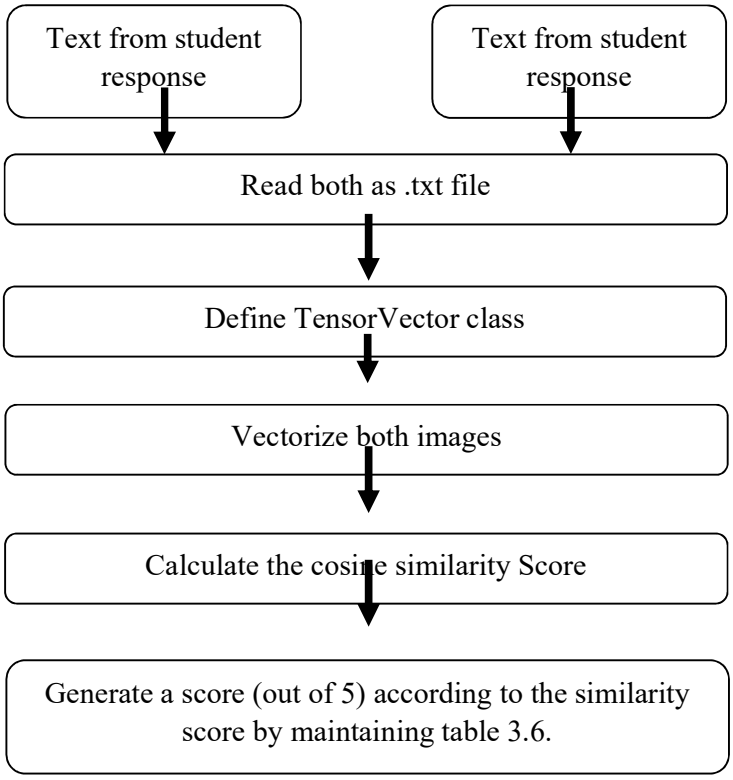


Figure 3.10: Workflow of cosine similarity for figure evaluation

Table 3.6: Scoring system for cosine similarity for figure

Similarity score	Score (out of 5)
>=90	5
>=85	4
>=80	3
>=75	2
>=70	1

3.6.2.2 Using Euclidean distance

Euclidean distance is a commonly used measure of distance between two points in Euclidean space. In image processing, Euclidean distance can be used as a measure of similarity between two images.

To calculate the Euclidean distance between two images, one first needs to represent the images as vectors. This can be done by flattening the pixel values of each image into a one-dimensional array. The Euclidean distance between the two vectors is then calculated as the square root of the sum of the squared differences between the corresponding elements of the vectors.

When comparing two images using Euclidean distance, a smaller distance indicates a greater degree of similarity between the two images. This can be useful in applications such as image retrieval, where one wants to retrieve images that are similar to a given query image.

However, Euclidean distance has some limitations when it comes to image similarity. One issue is that it does not take into account the spatial relationship between pixels in the images. Two images with very different spatial arrangements of similar pixels may have a high Euclidean distance, even though they are visually similar. Additionally, Euclidean distance may be sensitive to changes in lighting, contrast, and other factors that affect the pixel values of an image.

The formula for Euclidean distance between two n-dimensional vectors x and y is:

$$d(x, y) = \text{sqrt}((x1 - y1)^2 + (x2 - y2)^2 + ... + (xn - yn)^2) \quad (3.5)$$

where x1, x2, ..., xn and y1, y2, ..., yn are the corresponding elements of the two vectors.

In the context of image similarity, x and y represent the pixel values of the two images being compared, which are flattened into one-dimensional arrays before calculating the Euclidean distance.

This method includes the following steps:

- Store the figure image from the student answer sheet as a .jpg file.
- Store the expected Figure image provided by the teacher as a .jpg file.
- Read both of them as a .jpg file.
- Apply Canny edge detection to extract edges.
- Find contours in the image
- Calculate the Hu moments for each contour
- Calculate Euclidean distance between two the Hu moments
- Calculate the correlation and mean square error between the two images
- Calculate the similarity between the image according to the formula.
- Generate a score (out of 5) according to the similarity score by maintaining table 3.7.

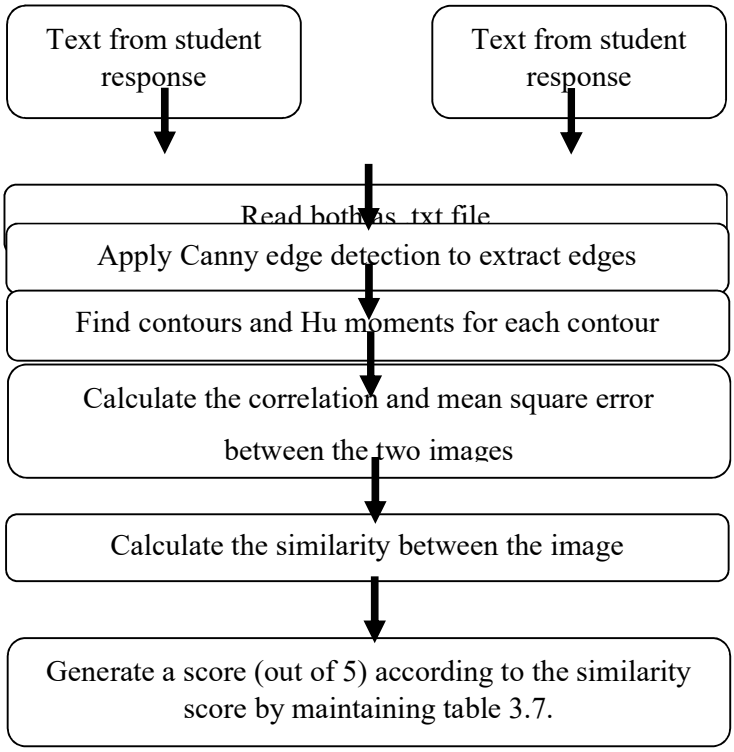


Figure 3.11: Workflow of Euclidean distance

Table 3.7: Scoring system for Euclidean distance

Similarity score	Score (out of 5)
>=90	5
>=85	4
>=80	3
>=75	2
>=70	1

Overall, the methodology involves collecting relevant data, pre-processing the data to ensure suitability, using appropriate techniques to analyze the data, and evaluating the results based on the objectives of the thesis.

CHAPTER 4

IMPLEMENTATION AND ANALYSIS

4.1 INTRODUCTION

The result analysis section of a thesis is a critical part of the research process. It is where the findings of the study are presented, and the implications of those findings are discussed. In this section, the researcher interprets the data collected during the study and draws conclusions based on the findings. The purpose of this section is to provide a detailed account of the results obtained and to evaluate the effectiveness of the research methodology employed. The results are presented in a clear and concise manner, and the researcher must demonstrate the significance of the findings and their contribution to the field of study.

The main objective of this thesis is to demonstrate a model of the automatic answer evaluation system. As in the section, there are multiple techniques, it is difficult to choose the best one for each section. So, here trying to find the best techniques by comparing them with each other.

4.2 TEXT EVALUATION

The text evaluation process using the proposed technique yielded promising results. The use of pre-processing techniques such as punctuation removal and text alignment helped in improving the accuracy of the model. The writing score calculated using Grammarly also provided valuable insights into the quality of the student's response. The cosine similarity and NLP-based methods for calculating the similarity score between the student's response and the expected answer provided a comprehensive evaluation of the

text. The results showed that the NLP-based method outperformed the cosine similarity method in terms of accuracy.

Now moving to the accuracy analysis section. The method used in text evaluation includes:

- Keyword matching score
- Similarity matching score, and
    - cosine similarity, and
    - NLP-based cosine similarity.
- Plagiarism check score

After performing the text evaluation using our proposed technique, we obtained the similarity scores for each student's answer compared to the expected answer. The results showed a range of similarity scores, with some answers being highly similar to the expected answer and others being vastly different.

Now, in this Section, take 10 different students' responses to a specific question and show an analysis of the performance of each individual technique and their average performance.

4.2.1 Samples

Question:

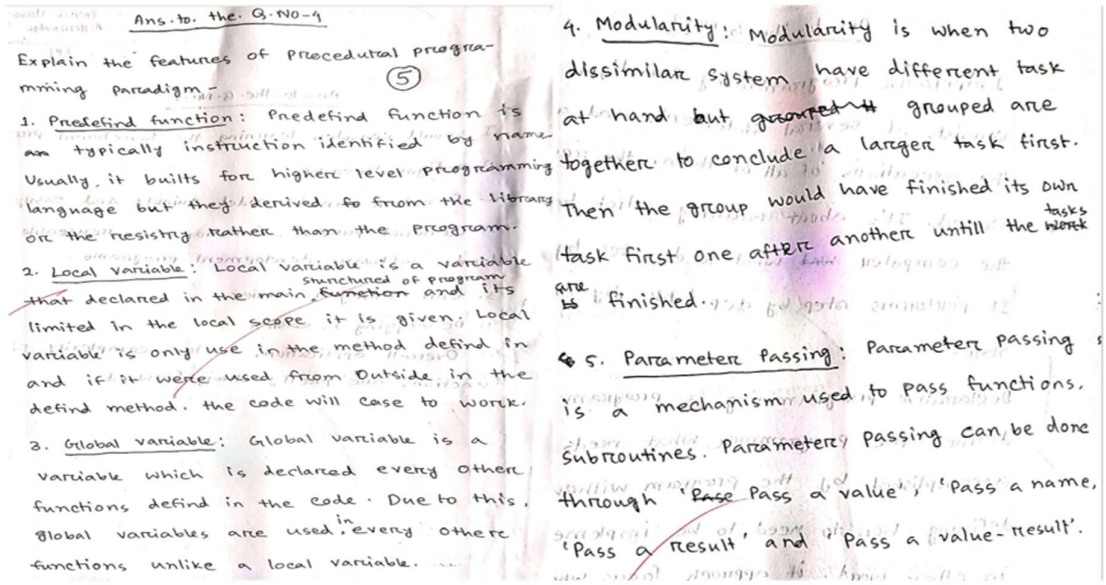
Table 4.1: Sample question 1

Sl no.	Question	Mark
1.	Explain the features of the procedural programming paradigm	5

Answers in the scripts

Here is a sample of students' answer scripts:

Sample 1



Ans. to the Q. NO-4

Explain the features of procedural programming paradigm. (5)

1. Predefind function: Predefind function is an typically instruction identified by name. Usually, it builts for higher level programming language but they derived fo from the library or the resistry rather than the program.

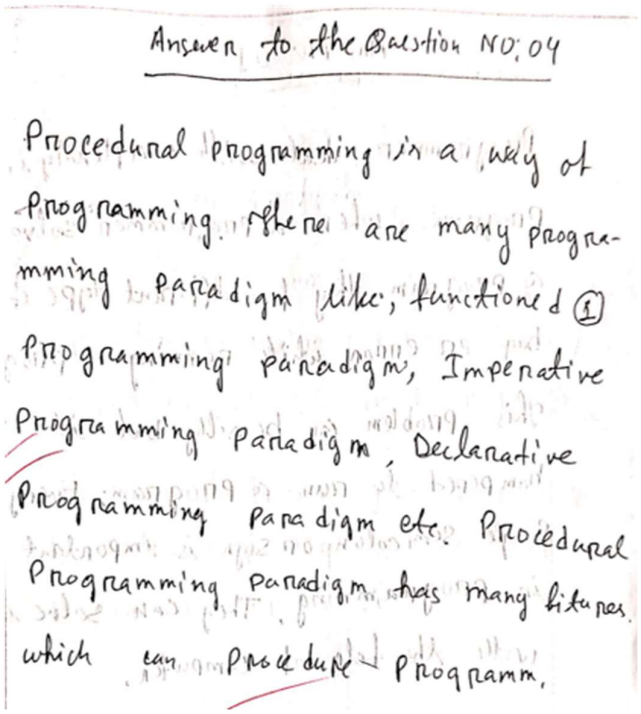
2. Local variable: Local variable is a variable that declared in the main function and its limited in the local scope it is given. Local variable is only use in the method defind in and if it were used from Outside in the defind method. the code will case to work.

3. Global variable: Global variable is a variable which is declared every other functions defind in the code. Due to this, global variables are used in every others functions unlike a local variable.

4. Modularity: Modularity is when two dissimilar system have different task at hand but grouped grouped are together to conclude a larger task first. Then the group would have finished its own task first one after another untill the tasks are finished.

5. Parameter passing: Parameter passing is a mechanism used to pass functions. Subroutines. Parameter passing can be done through 'Pass a value', 'Pass a name, 'Pass a result' and 'Pass a value-result'.

Sample 2



Answer to the Question NO: 04

Procedural programming is a way of Programming. There are many programming paradigm like, functioned (1) Programming paradigm, Imperative Programming Paradigm, Declarative Programming Paradigm etc. Procedural Programming Paradigm has many fetures which can Procedure Programm.

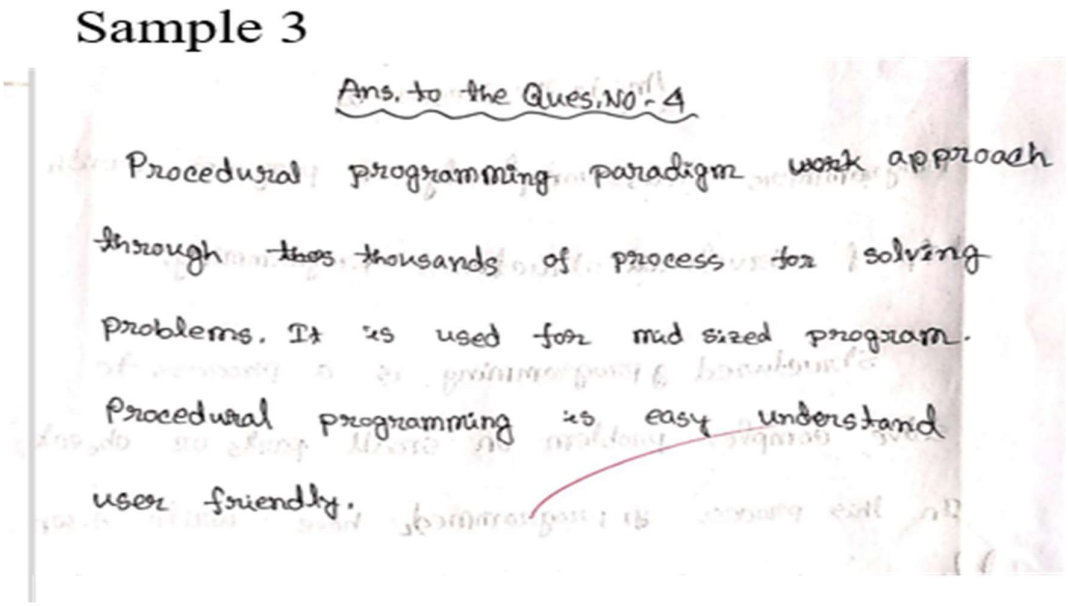


Figure 4.1: Answers in the sample scripts for text evaluation

4.2.2 First Case

In this category, considering writing score, keyword matching score, similarity score by cosine similarity, and plagiarism check score.

Cosine similarity is a widely used technique to measure the similarity between two vectors of textual data. In our study, we used cosine similarity to measure the similarity between the text extracted from the student's answer sheet and the expected answer sheet. To analyze the accuracy of this technique, we randomly selected a set of answer sheets from the dataset and calculated the cosine similarity score between the student's answer and the expected answer. We then manually evaluated the similarity between the two answers and compared it with the cosine similarity score obtained.

Table 4.2. Evaluated mark for first case

Sl	Keyword matching		Writing score		Similarity with cosine similarity		Plagiarism		Final obtained mark (out of 5)
	Matching (%)	Mark	Writing score	Mark	Matching (%)	Mark	Matching (%)	Mark	
1	33	2	38	2	13	2	2	1	1.75
2	72	5	45	2	30	4	4	3	3.5
3	18	1	68	4	12	2	2	1	2
4	68	4	53	3	35	4	5	4	3.75
5	76	5	62	4	40	5	9	5	4.75
6	9	0	68	4	14	2	2	1	1.75
7	9	0	50	3	10	2	17	5	2.5
8	82	5	41	2	43	5	11	5	4.25
9	63	4	54	3	21	3	7	5	3.75
10	8	0	51	3	3	0	2	1	1

4.2.3 Second Case

In this category, considering writing score, keyword matching score, NLP-based similarity score by cosine similarity, and plagiarism check score.

The NLP-based cosine similarity technique showed a significant improvement in text evaluation accuracy compared to the basic cosine similarity technique. This technique uses NLP algorithms to analyze the meaning of the text instead of just comparing the frequency of words, resulting in a more accurate evaluation.

Table 4.3 Evaluated mark for second case

Id	Keyword matching		Writing score		Similarity with NLP-based cosine similarity		Plagiarism		Final obtained mark (out of 5)
	Matching (%)	Mark	Writing score	Mark	Matching (%)	Mark	Matching (%)	Mark	
1	33	2	38	2	86	2	2	1	1.75
2	72	5	45	2	91	5	4	3	3.75
3	18	1	68	4	85	1	2	1	1.75
4	68	4	53	3	90	4	5	4	3.75
5	76	5	62	4	92	5	9	5	4.75
6	9	0	68	4	83	0	2	1	1.5
7	9	0	50	3	86	2	17	5	2.5
8	82	5	41	2	91	5	11	5	4.25
9	63	4	54	3	87	3	7	5	3.75
10	8	0	51	3	85	1	2	1	1.25

4.2.4 Accuracy Testing

In this section accuracy of text evaluation for both cases will be tested. For proper analysis suitable tables and graphs are included.

Table 4.4: Accuracy analysis of both case

Sl no.	Evaluated mark as per first case (Out of 5)	Evaluated mark as per second case (Out of 5)	Examiner's mark (Out of 5)	Accuracy for first case (%)	Accuracy for second case (%)
1	1.75	1.75	2	95	95
2	3.5	3.75	5	70	75
3	2	1.75	0.5	70	75
4	3.75	3.75	4	95	95
5	4.75	4.75	5	95	95
6	1.75	1.5	1	85	90
7	2.5	2.5	0.5	60	60

8	4.25	4.25	5	85	85
9	3.75	3.75	3.5	95	95
10	1	1.25	2	80	85

Average accuracy for the first case is 83% and for second case is 85%. Here second case perform little better than the first case. Now discussing both case by some graph analysis.

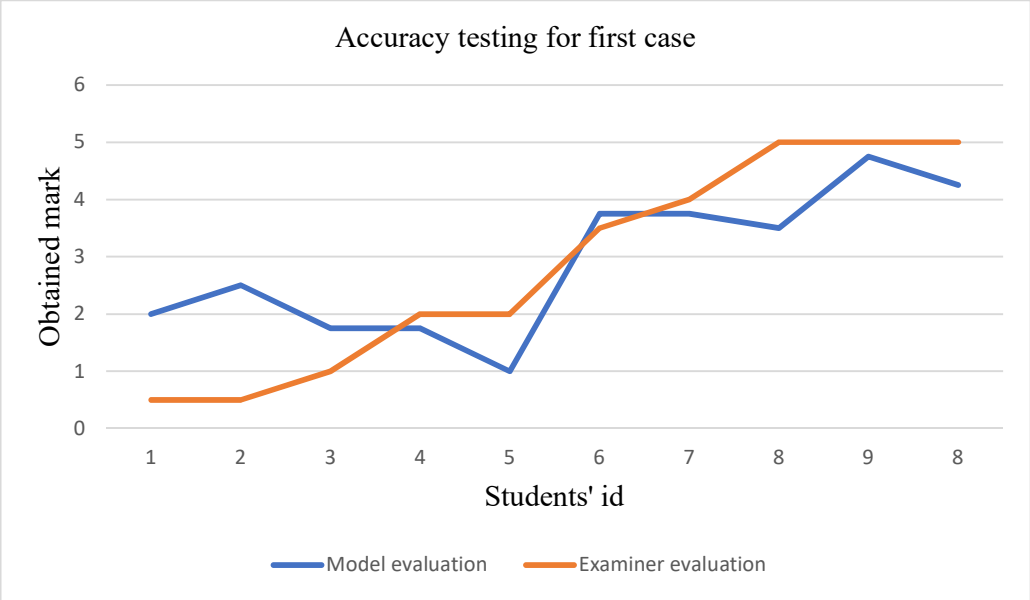


Figure 4.2: Accuracy testing for first case

This line graph demonstrates that there are some minor discrepancies between the examiner's evaluation and the model evaluation at certain spots. And there are some significant gaps. Hence, this process exhibits two different sorts of behavior. Yet, the first case's approach offers 83% accuracy.

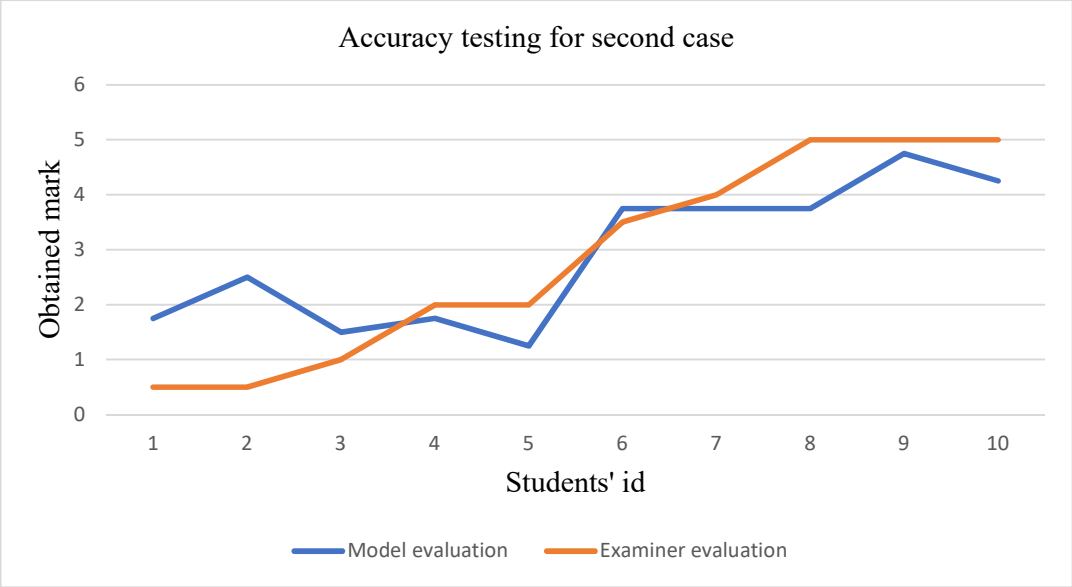


Figure 4.3: Accuracy testing for second case

This line graph shows that there are a few small differences between the model evaluation and the examiner's evaluation at specific locations. Moreover, there are several big gaps. As a result, this process displays two distinct types of behavior. Yet, the method used in the second case is 85% accurate.

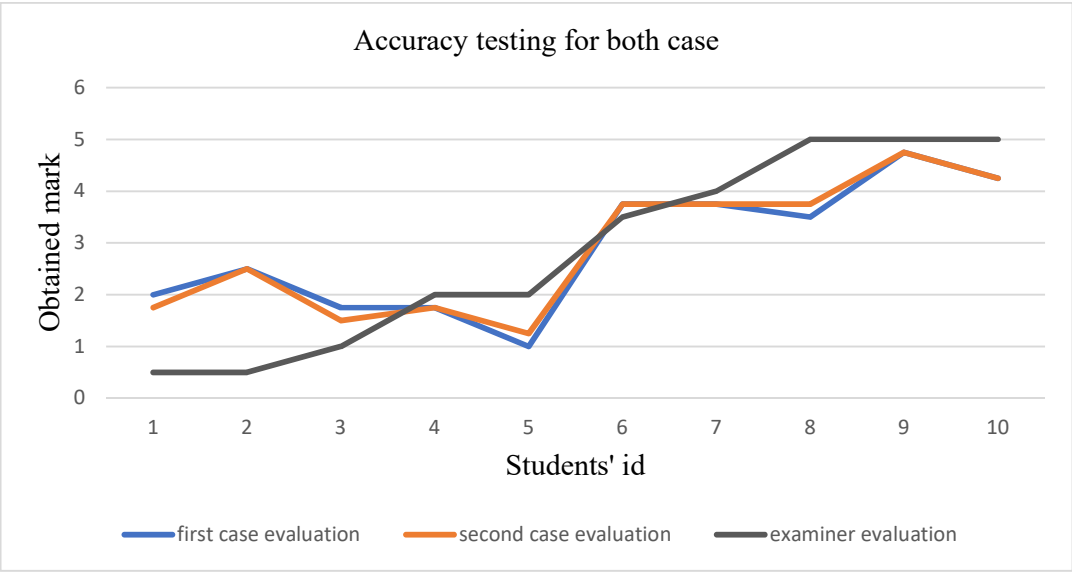


Figure 4.4: Accuracy testing for both case

Both graphs depict the evaluation process and indicate that there are differences between the model evaluation and the examiner's evaluation. In both cases, there are minor discrepancies and significant gaps between the two evaluations at certain locations. Additionally, both graphs suggest that there are two distinct types of behavior in the evaluation process.

The primary difference between the two graphs is the accuracy rate mentioned for the first situation. The first graph states that the method used in the first situation is 85% accurate, while the second graph mentions 83% accuracy for the same situation. The discrepancy in accuracy rate could be due to differences in the methods used to calculate accuracy or due to rounding.

4.3 FIGURE EVALUATION

The use of these two methods for figure evaluation shows varying degrees of accuracy. While both techniques are able to evaluate the figure similarity between student response and expected answer, there are some limitations to each method. This method includes:

- Cosine Similarity
- Euclidean Distance

Now, in this Section, take 10 different students' responses to a specific question and show an analysis of the performance of each individual technique.

4.3.1 Samples

Question:

Table 4.5: Sample question 2

Sl no.	Question	Mark
1.	Draw a figure of center-tap full wave rectifier	4

Student Response:

Here is a sample of student responses:

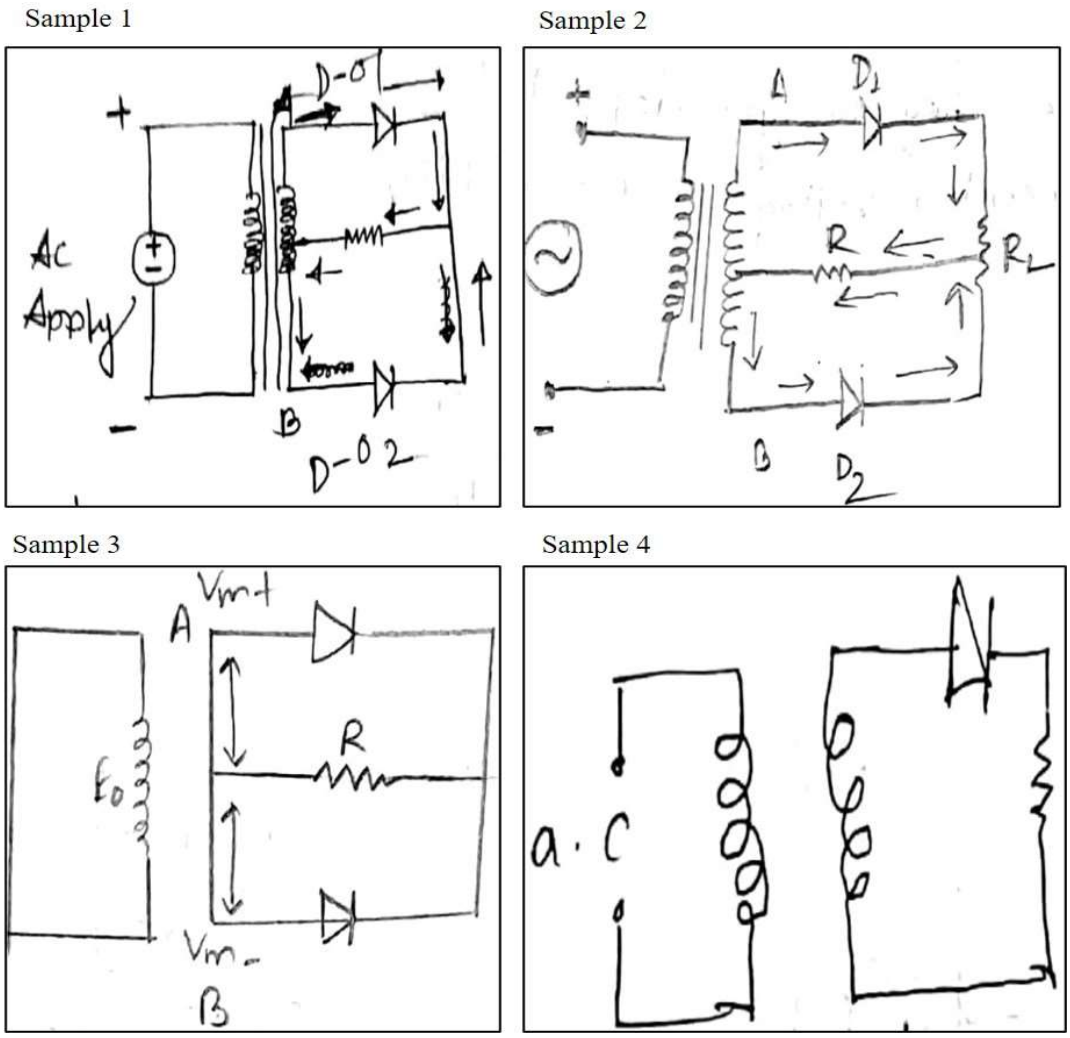


Figure 4.5: Answers in the sample scripts for figure evaluation

4.3.2 Cosine Similarity

Cosine similarity is a commonly used technique to measure the similarity between two vectors in NLP and image processing. In this thesis, cosine similarity was applied to evaluate the similarity between the expected answer and student response for figure evaluation.

Table 4.6: Evaluated marks for cosine similarity

Id	Cosine Similarity		Final obtained mark (out of 5)
	Matching (%)	mark (out of 5)	
1	89	4	3.2
2	84	3	2.4
3	94	5	4
4	89	4	3.2
5	75	2	1.6
6	88	4	3.2
7	90	5	4
8	84	3	2.4
9	79	2	1.6
10	68	0	0

4.3.3 Euclidean Distance

Euclidean Distance is another method used for figure evaluation in this thesis. The accuracy of this method is analyzed using the same dataset as in cosine similarity. This method shows less stability and consistency compared to cosine similarity, with more spikes and dips in the accuracy graph.

Table 4.7: Evaluated marks Euclidean Distance

Roll	Euclidean Distance		Final obtained mark (out of 4)
	Matching (%)	Mark (out of 5)	
1	3.2	2.5	2
2	2	4	3.2
3	0.6	5	4
4	3.7	2.5	2
5	0.2	5	4
6	0.4	5	4

7	1.2	4	3.2
8	0.26	5	4
9	45	0	0
10	52	0	0

4.3.4 Accuracy testing

In this section accuracy of figure evaluation for both techniques will be tested. For proper analysis suitable tables and graphs are included.

Table 4.8: Accuracy testing of both techniques

Sl no.	Evaluated mark as per Cosine similarity (Out of 5)	Evaluated mark as per Euclidean distance (Out of 5)	Examiner's mark (Out of 4)	Accuracy for Cosine similarity (%)	Accuracy for Euclidean distance (%)
1	3.2	2	4	80	50
2	2.4	3.2	2	90	70
3	4	4	4	100	100
4	3.2	2	3	95	75
5	1.6	4	1	85	25
6	3.2	4	4	80	100
7	4	3.2	4	100	80
8	2.4	4	1	65	25
9	1.6	0	0	60	100
10	0	0	0	100	100

Average accuracy of cosine similarity for figure evaluation is about 83% and average accuracy for Euclidean distance is about 72%.

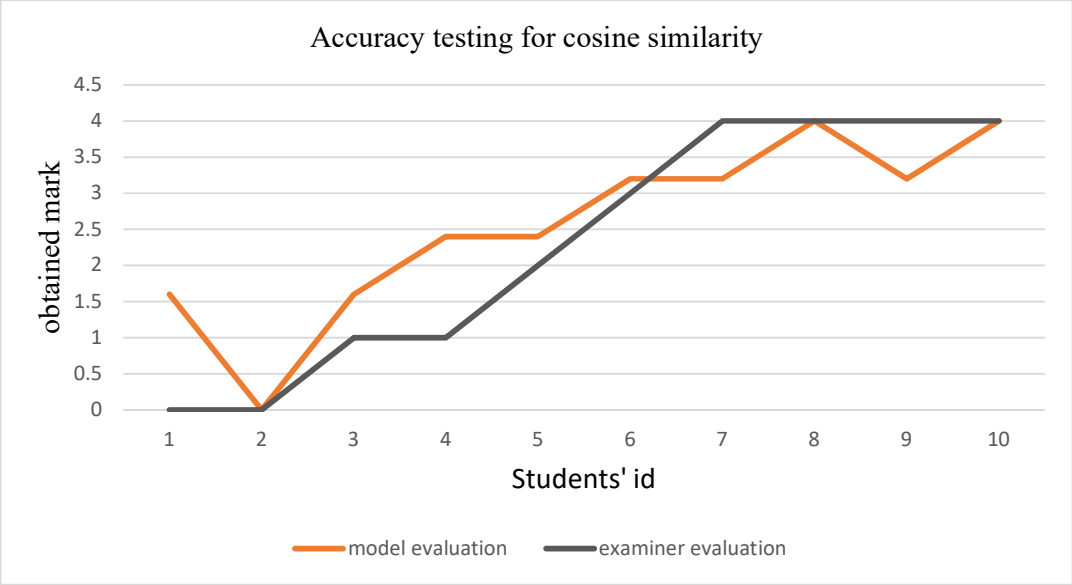


Figure 4.6: Accuracy testing for cosine similarity

This line graph demonstrates that, at some points, there are a few minor discrepancies between the model evaluation and the examiner's evaluation. There are also quite a few significant gaps. This process consequently exhibits two different forms of behavior. Yet, the first situation's technique is 85% accurate.

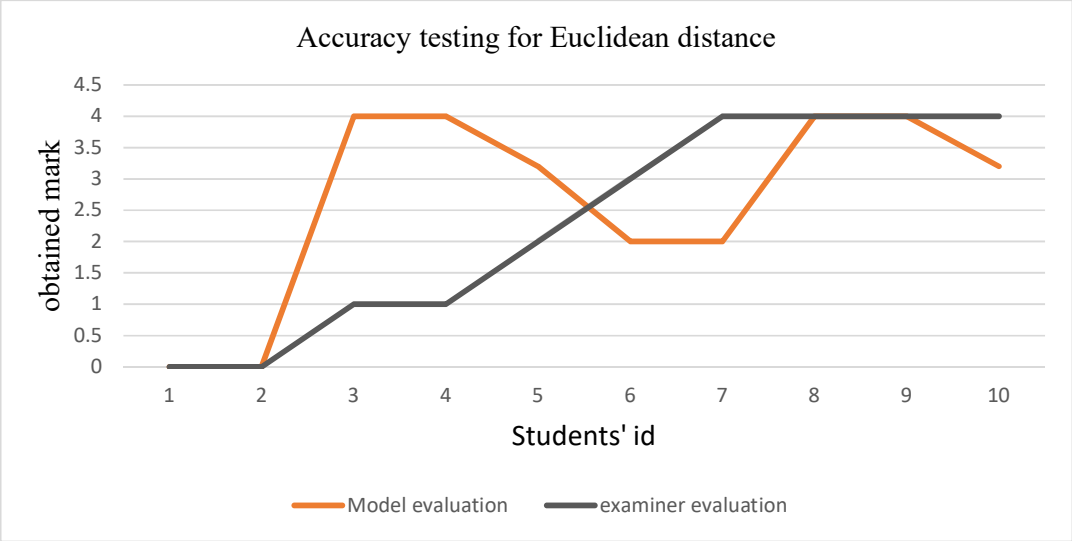


Figure 4.7: Accuracy testing for Euclidean distance

Large spaces exist between several of the lines. The closest points to one another are sparse. With 72% accuracy, we may claim that this technique has significant shortcomings.

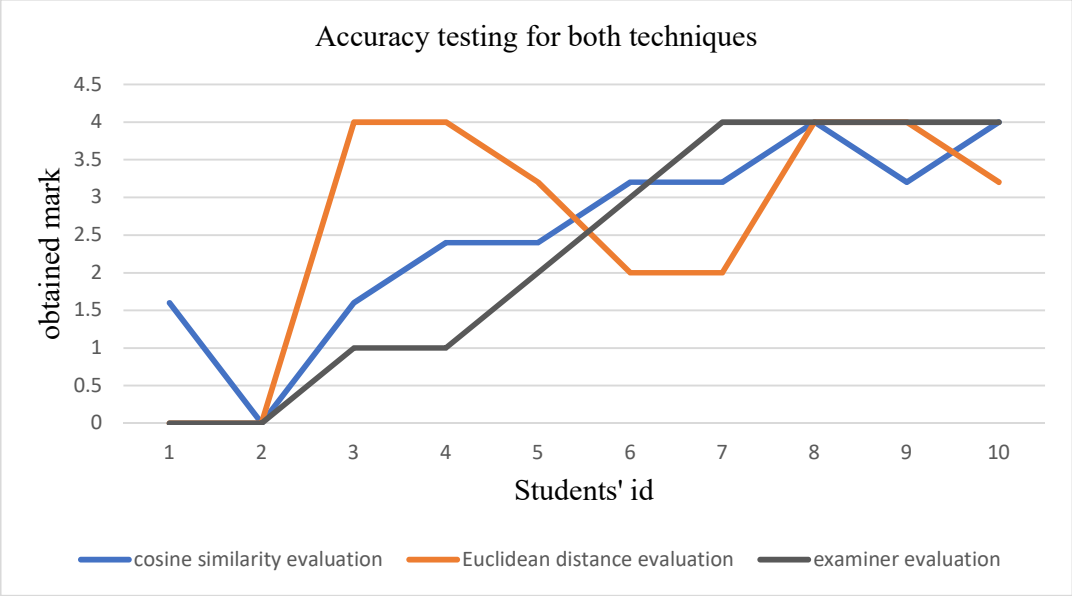


Figure 4.8: Accuracy testing for both techniques

The line of cosine similarity evaluation shows a more stable and consistent accuracy compared to the Euclidean distance. While the first line exhibits a few minor discrepancies and significant gaps, it still achieves an overall accuracy of 85%. On the other hand, the second line has larger spaces between the lines, indicating greater discrepancies in accuracy. Despite achieving 72% accuracy, this technique still has significant shortcomings. Therefore, it can be suggested that the first technique may be more reliable and consistent in evaluating text answers, while the second technique needs further improvement to increase its accuracy and reduce the discrepancies.

4.4 DISCUSSION

The results of our study indicate that our computer-based answer script evaluation system is effective in evaluating both text and figures in answer scripts. The accuracy scores obtained for both text and figure evaluation demonstrate the potential for our system to reduce the workload of educators while providing timely feedback to students.

In text evaluation, we considered two cases and used different methods for each case. For the first case, we achieved an accuracy score of 83% using writing score, keyword matching score, cosine similarity score, and plagiarism matching score. For the second case, we achieved an accuracy score of 85% using writing score, keyword matching score, NLP-based cosine similarity score, and plagiarism matching score. These results suggest that the use of NLP-based cosine similarity score in the second case improved the accuracy of the evaluation compared to the cosine similarity score used in the first case.

In figure evaluation, we used cosine similarity and Euclidean distance techniques. Our system achieved an accuracy score of 85% using the cosine similarity technique, while the accuracy score was 72% using the Euclidean distance technique. These results suggest that the cosine similarity technique is more effective than the Euclidean distance technique for figure evaluation.

Overall, the results of our study demonstrate the potential of our computer-based answer script evaluation system to provide effective and reliable evaluation of both text and figures in answer scripts. However, there is still room for improvement, and future work could focus on further enhancing the accuracy of the system by incorporating additional evaluation techniques or refining the existing ones.

CHAPTER – 5

CONCLUSION AND FUTURE WORK

5.1 INTRODUCTION

The conclusion chapter is the final section of this thesis, where the results, discussions, and research objectives are summarized. The conclusion chapter provides an overview of the entire study, highlighting the major contributions, limitations, and future research directions. In this chapter, the findings of the study are synthesized to address the research questions and objectives, and the significance of the results is discussed. The conclusion chapter also provides an opportunity to reflect on the challenges and opportunities encountered during the research process, and to suggest areas for future research. This chapter aims to draw a comprehensive conclusion from the results obtained and to contribute to the advancement of the field of education technology.

5.2 CONCLUSION

In this thesis, we proposed a computer-based answer script evaluation system that effectively evaluates both text and figures in answer scripts. Our system uses a combination of different evaluation techniques, including writing score, keyword matching score, NLP-based cosine similarity score, plagiarism matching score, cosine similarity, and Euclidean distance, to achieve high accuracy scores.

Our study demonstrated that our system has the potential to greatly reduce the workload of educators while providing timely feedback to students. The accuracy scores obtained for both text and figure evaluation suggest that our system is effective in evaluating answer scripts, and that the use of different evaluation techniques can further improve the accuracy of the evaluation.

Overall, our system represents a significant step forward in the field of automated answer script evaluation. Future work could focus on further enhancing the accuracy of the system by incorporating additional evaluation techniques or refining the existing ones. With the continued development of automated evaluation systems, we can expect to see increased efficiency and accuracy in the evaluation of answer scripts, ultimately improving the quality of education.

5.3 FUTURE WORK SCOPE

The future work scope for this topic includes further improving the accuracy and efficiency of the proposed model. One possible direction for future research is to explore more advanced deep learning techniques, such as recurrent neural networks (RNNs) and transformers, which have shown promising results in various natural language processing tasks. Additionally, the model could be extended to handle different languages and writing systems, allowing for broader applicability in diverse educational settings. Another potential avenue for future work is to investigate the use of the model in other areas of education, such as automated grading systems and personalized learning platforms. Finally, it is important to consider ethical considerations related to the use of such technologies in education, such as privacy concerns and potential biases, and to develop appropriate safeguards to ensure that the model is used in a fair and responsible manner.

Working on CNN Model with a large enough dataset may be another direction. Use other Similarity techniques to generate better accuracy. Evaluate Mathematical expressions and problems can be a concern for future work.

REFERENCES

1. X. Hu and H. Xia, ‘‘Automated assessment system for subjective questions based on LSI,’’ in Proc. 3rd Int. Symp. Intell. Inf. Technol. Secure. Information., Apr. 2010, pp. 250–254.
2. M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, ‘‘From word embeddings to document distances,’’ in Proc. Int. Conf. Mach. Learn., 2015, pp. 957–966.
3. J. E. Kim, K. Park, J. M. Chae, H. J. Jang, B. W. Kim, and S. Y. Jung, ‘‘Automatic scoring system for short descriptive answer written in Korean using Lexico-semantic pattern,’’ Soft Comput., vol. 22, no. 13, pp. 4241–4249, 2018.
4. M. Oghbaie and M. M. Zanjireh, ‘‘Pairwise document similarity measure based on present term set,’’ J. Big Data, vol. 5, no. 1, pp. 1–23, Dec. 2018.
5. Orkphol and W. Yang, ‘‘Word sense disambiguation using cosine similarity collaborate with Word2vec and WordNet,’’ Future Internet, vol. 11, no. 5, p. 114, May 2019.
6. C. Xia, T. He, W. Li, Z. Qin, and Z. Zou, ‘‘Similarity analysis of law documents based on Word2vec,’’ in Proc. IEEE 19th Int. Conf. Softw. Qual., Rel. Secure. Companion (QRS-C), Jul. 2019, pp. 354–357.
7. R. S. Wagh and D. Anand, ‘‘Legal document similarity: A multicriteria decision-making perspective,’’ PeerJ Comput. Sci., vol. 6, p. e262, Mar. 2020.
8. M. Alian and A. Awajan, ‘‘Factors affecting sentence similarity and paraphrasing identification,’’ Int. J. Speech Technol., vol. 23, no. 4, pp. 851–859, Dec. 2020.
9. J. Muangprathub, S. Kajornkasirat, and A. Wanichsombat, ‘‘Document plagiarism detection using a new concept similarity in formal concept analysis,’’ J. Appl. Math., vol. 2021, pp. 1–10, Mar. 2021.
10. G. Jain and D. K. Lobiyal, ‘‘Conceptual graphs based approach for subjective answers evaluation,’’ Int. J. Conceptual Struct. Smart Appl., vol. 5, no. 2, pp. 1–21, Jul. 2017.
11. V. Bahel and A. Thomas, ‘‘Text similarity analysis for evaluation of descriptive answers,’’ 2021, arXiv:2105.02935.

12. Sheeba Praveen, “An Approach to Evaluate Subjective Questions for Online Examination System”, Assistant Professor, Dept. CSE, Integral University, Lucknow, U.P, India.
13. B Vanni, M. shyni, and R. Deepalakshmi, ”High accuracy optical character recognition algorithms using learning an array of ANN” in Proc. 2014 IEEE International Conference on Circuit, Power and Computing Technologies (ICCPCT), 2014 International Conference.
14. An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition ”, arXiv:1507.05717v1 [cs.CV] 21 Jul 2015.
15. Yusuf Perwej, Ashish Chaturvedi, “ Neural Networks for Handwritten English Alphabet Recognition ”, International Journal of Computer Applications (0975 – 8887) Volume 20– No.7, April 2011.
16. J.Pradeep , E.Srinivasan and S.Himavathi, “Neural network based handwritten character recognition system without feature extraction ”, International Conference on Computer, Communication and Electrical Technology – ICCCET 2011, 18th & 19th March 2011.
17. K Arlitsch, J Herbert “Microfilm, paper, and OCR: Issues in newspaper digitization. the Utah digital newspapers program” Microform & Imaging Review, 2004 - degruyter.com
18. Shai Shalev-Shwartz and Shai Ben-David “Understanding Machine Learning: From Theory to Algorithms”, ©2014 [11] George Nagy, Stephen V. Rice, and Thomas A. Nartker, “Optical Character Recognition: An Illustrated Guide to the Frontier (The Springer International Series in Engineering and Computer Science)”
19. Batuhan Balci, Dan Saadati, Dan Shiferaw, “Handwritten Text Recognition using Deep Learning ”, CS231n: Convolutional Neural Networks for Visual Recognition Spring 2017 project report.
20. Judy McKimm, Carol Jollie, Peter Cantillon, “ABC of learning and teaching Web based learning” http://www.bmj.com/
21. Yuan, Zhenming, et al. A Web-Based Examination and Evaluation System for Computer Education. Washington, DC: IEEE Computer Society, 2006.

22. Effie Lai-Chong Law, et al. Mixed-Method Validation of Pedagogical Concepts for an Intercultural Online Learning Environment. New York: Association for Computer Machinery, 2007.
23. Lan, Glover, et al. Online Annotation- Research and Practices. Oxford UK: Elsevier Science Ltd, 2007.
24. Sophal Chao and Dr. Y.B Reddy Online examination Fifth International Conference on Information Technology: New Generations, 2008
25. Hanumant R. Gite, C.Namrata Mahender "Representation of Model Answer: Online Subjective Examination System" National conference NC3IT2012 Sinhgad Institute of Computer Sciences Pandharpur.
26. "Discourse analysis" website http://www.tlumaczenia _angielski.info/linguistic/discourse.htm
27. H. Mittal and M. S. Devi, ''Subjective evaluation: A comparison of several statistical techniques,'' Appl. Artif. Intell., vol. 32, no. 1, pp. 85–95, Jan. 2018.
28. L. A. Cutrone and M. Chang, ''Automarking: Automatic assessment of open questions,'' in Proc. 10th IEEE Int. Conf. Adv. Learn. Technol., Sousse, Tunisia, Jul. 2010, pp. 143–147.
29. G. Srivastava, P. K. R. Maddikunta, and T. R. Gadekallu, ''A two-stage text feature selection algorithm for improving text classification,'' Tech. Rep., 2021.
30. H. Mangassarian and H. Artail, ''A general framework for subjective information extraction from unstructured English text,'' Data Knowl. Eng., vol. 62, no. 2, pp. 352–367, Aug. 2007.
31. B. Oral, E. Emekligil, S. Arslan, and G. Eryigit, ''Information extraction ˇ from text intensive and visually rich banking documents,'' Inf. Process. Manage., vol. 57, no. 6, Nov. 2020, Art. no. 102361.
32. H. Khan, M. U. Asghar, M. Z. Asghar, G. Srivastava, P. K. R. Maddikunta, and T. R. Gadekallu, ''Fake review classification using supervised machine learning,'' in Proc. Pattern Recognit. Int. Workshops Challenges (ICPR). New York, NY, USA: Springer, 2021, pp. 269–288.
33. S. Afzal, M. Asim, A. R. Javed, M. O. Beg, and T. Baker, ''URLdeepDetect: A deep learning approach for detecting malicious URLs using semantic vector models,'' J. Netw. Syst. Manage., vol. 29, no. 3, pp. 1–27, Mar. 2021.

34. N. Madnani and A. Cahill, ‘‘Automated scoring: Beyond natural language processing,’’ in Proc. 27th Int. Conf. Comput. Linguistics (COLING), E. M. Bender, L. Derczynski, and P. Isabelle, Eds. Santa Fe, NM, USA: Association for Computational Linguistics, Aug. 2018, pp. 1099–1109.
35. Z. Lin, H. Wang, and S. I. McClean, ‘‘Measuring tree similarity for natural language processing based information retrieval,’’ in Proc. Int. Conf. Appl. Natural Lang. Inf. Syst. (NLDB) (Lecture Notes in Computer Science), vol. 6177, C. J. Hopfe, Y. Rezgui, E. Métais, A. D. Preece, and H. Li, Eds. Cardiff, U.K.: Springer, 2010, pp. 13–23.
36. G. Grefenstette, ‘‘Tokenization,’’ in Syntactic Wordclass Tagging. Springer, 1999, pp. 117–133.
37. K. Sirts and K. Peekman, ‘‘Evaluating sentence segmentation and word Tokenization systems on Estonian web texts,’’ in Proc. 9th Int. Conf. Baltic (HLT) (Frontiers in Artificial Intelligence and Applications) vol. 328, U. Andrius, V. Jurgita, K. Jolantai, and K. Danguole, Eds. Kaunas, Lithuania: IOS Press, Sep. 2020, pp. 174–181.
38. A. Schofield, M. Magnusson, and D. M. Mimno, ‘‘Pulling out the stops: Rethinking stopword removal for topic models,’’ in Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics (EACL) vol. 2, M. Lapata, P. Blunsom, and A. Koller, Eds. Valencia, Spain: Association for Computational Linguistics, 2017, pp. 432–436.
39. M. Çagatayli and E. Çelebi, ‘‘The effect of stemming and stop-wordremoval on automatic text classification in Turkish language,’’ in Proc. 22nd Int. Conf. Neural Inf. Process. (ICONIP) (Lecture Notes in Computer Science), vol. 9489, S. Arik, T. Huang, W. K. Lai, and Q. Liu, Eds. Istanbul, Turkey: Springer, 2015, pp. 168–176.