Grenze International Journal of Engineering and Technology, June Issue

Automated Descriptive and Diagrammatic Answer Scoring Tool

Tanishka Shinde[1], Shruti Mehta[2], Shrutee Bothara[3], Shreya Bothara[4] and Prof. Minal Nerkar[5]
[1-5]All India Shri Shivaji Memorial Society's Institute of Information Technology/Computer Engineering, Pune, India
Email: tanishkashinde57@gmail.com, {shrutimehta1301, shruteebothara, shreyabothara04}@gmail.com ,
{minal.nerkar}@aissmsioit.org

***Abstract*—Examinations are a significant phase when it comes to scrutinizing the intellectual abilities of the examinee. One is being tested for their knowledge in various domains. This stage is not just crucial for the student, but also for the examiner. The examiner needs to be proficient in his/her domain. Manual paper checking is the norm nowadays, but in the era of automation and technology, one needs to elevate the quality and process of paper checking and stay plugged in with increasing automation. This is also because manual paper checking can be tedious, time consuming and various factors of prejudice can prove unfair for the student. Not to forget grading mistakes that occur unknowingly by the examiner. We have brought forth An Automated Descriptive and Diagrammatic Answer Scoring Tool to upgrade paper checking procedures. This will help reduce bias time consumption, stress and will not just calculate descriptive answers, but also evaluate diagrams and mathematical formulae. It will also take into account, Bloom's Taxonomy, while grading the answers. For descriptive answer evaluation this model uses various algorithms whose collective accuracy comes up to 98%. Whereas, diagrammatical and mathematical answer evaluation gives an accuracy of 95% and 96% respectively.**

***Index Terms*— Descriptive Answer Evaluation, Natural Language Processing (NLP), Text Extraction, Concept Graph, Image Processing, Mathematical Equation Processing, Fuzzy String Matching, Ontological Learning, Diagrammatical Evaluation, Convolutional Neural Network (CNN).**

I. INTRODUCTION

Examinations are a challenging phase, not just for the students but also for teachers. While students are being tested for their intellect, reasoning and academic under- standing, teachers are also indirectly being tested for their efficiency to assess students and understanding of the particular subject they're assessing. Each teacher appointed for a subject needs to be adept and proficient in their particular discipline. This is usually done by means of manual paper checking. Manual paper checking has time and again proven to be a tedious and error prone process. It involves grading different types of answers, be it definitions, answers in brief, descriptive answers, diagrams and mathematical concepts. Furthermore, only technical keywords cannot cover a student's all-roundedness. One also has to be careful and tend to the student's language of answering. This is where grammatical concepts and fine use of vocabulary comes into action. There is an abundance of examples where teachers have haphazardly overlooked grammar, totalling of marks and other minor flaws done by students. Additionally, bias factor between the teacher and student can be a major flaw. Time constraints for manual paper

Grenze ID: 01.GIJET.9.2.567

checking have also proved exhausting for teachers. Since there is minimal interaction between teachers and students, it is confusing for students to understand the areas they lack in and progress tracking becomes complicated. Ultimately, all the systems and models that exist to date, have not clarified the concepts of diagrammatical and mathematical formulae evaluation Keeping in mind all these factors, we have proposed An Automated Descriptive and Diagrammatic Answer Scoring tool that will tend to these flaws present in manual paper checking and elevate the quality of student's paper assessment.

II. LITERATURE SURVEY

Reference [1] proposes an approach of keyword matching. Keywords written by the student are pulled out and compared to expected keywords. Cosine similarity, Term Frequency Inverse Document Frequency (TF-IDF) are some methods which are found to be implemented in this specific system. Although, there is no facility of providing partial scores, it remains as a future scope for their system. Ref [2] describes their implemented system which makes use of similarity checking using Synset. It focuses on linguistic features by pre-processing the answers using Natural Language Processing (NLP). This is evaluated using cosine similarity. They have also used dimensionality reduction for reducing the dimensions of vector space. Performance is measured using MSR paraphrase corpus and Li's benchmark datasets. Ref [3] has used Deep Learning and Neural Networks to grade descriptive answer scripts. This system is an amalgamation of NLP and Machine Learning and consists of three stages – Pre-processing, Semantic extraction and classification and grading. The student answers are converted into glove vector index and the model learning occurs through Recurrent Neural Network and Long Short Term Memory (LSTM). Lastly, based on semantic representation of answers taken from the LSTM layer, one hotted score is predicted. Ref [4] makes use of NLP text summarization techniques for pre-processing, also known as keyword- based text summarization. Weight value is assigned to each parameter to score marks. Various similarity measures such as Cosine, Bigram, Jaccard and Synonym similarity are used to check similarities between student answer and the expected answer and also as a parameter for final score. Ref [5] has focused on a mobile application to correct handwritten descriptive answers. Rapid Automatic Keyword Extraction (RAKE) algorithm extracts keywords and key phrases which are then checked for their similarity using Word Mover's Distance. A word2vec model is then used to embed these keywords and phrases that helps in answer evaluation. Although there are some flaws that exist in text extraction from handwritten answers. Ref [6] proposes an approach that makes use of Machine Learning and NLP techniques for answer evaluation. Various similarity measures such as Wordnet, Word2vec,Word Movers Frequency and multinomial Naive Bayes are used. The resulting accuracy of this model turns out to be 88% whereas in future the model can be trained more extensively for answers of a particular domain. Ref [7] has used Cosine-Similarity to match features and inspect Descriptive Answers but the overall accuracy is around 80-90%. This model proves to be less robust for checking Grammar and does not include Diagrams and Mathematical Equation Evaluation.

In Ref [8] methods are categorized into different classes which provide end-to-end approaches. This highlights the implementation of handwritten math recognition on various platforms and how different methods change the UI designs. When Ref [9] is summarized, the model heavily makes use of OCR and NLP techniques. Cosine similarity is put into action for grading marks. Ref [11] has implemented a model which makes use of self supervised, semantic analysis and feature based classification techniques. This model pulls off high precision for assessing handwritten answers, with an F1 score of 0.85.

In Ref [12] the system classifies images based on Deep Neural Network through TensorFlow. In consequence this Image processing technique gives 90% accuracy unlike our system which processes images with greater accuracy. Ref [13] devised an evaluation system that outputs a CSV file that contains final marks obtained by the examinee, although this system does not consider Mathematical Formulae Evaluation. Ref [14] worked on Mathematical Formulae Analysis using Vision Transformer and variety of functions such as SoftMax using Neural Network Decoder. This works on complex Mathematical Equations with decent accuracy. Ref [15] implements a system that recognizes Mathematical Equations using Support Vector Machine and K-Nearest Neighbours Algorithm, where the accuracies using both algorithms are 88.5% and 78.4% respectively. In Ref [16] the model relies on classifying mathematical formulae with the help of back- propagation neural network, K- nearest neighbour and Support Vector Machine Algorithms. An important feature of math-talk for visually impaired population has been introduced here. In Ref [17] the evaluation system makes use of TF-IDF and techniques like Self Organizing Map which our system has modified optimally using Concept Graph.

Ref [18], [19] in their work recognized handwritten character using Machine Learning and Neural Network but the accuracy of the system depends upon the dataset, requires human intervention and only evaluates descriptive answers unlike our system. Ref [20] devised a system that only considers descriptive answers in Bangla language. This is a major flaw that does not consider world’s most spoken language, English. Ref [21] implements a system that evaluates answers using Concept Graph and Sub-Graph but this system is also limited to descriptive answers. In Ref [22] and [25] the Answer Evaluation System makes use of Optical Character Recognition to evaluate theoretical answers. Although in Ref [22] the system evaluates only single answer instead of complete paper and Diagrammatical Evaluation is not considered. Ref [10] and [23] put into action an Answer Evaluation System that makes use of Optical Character Recognition and Convolutional Neural Network where model in Ref [23] achieves an accuracy of 92.86% unlike our proposed system that achieves an accuracy of 97.4% and model in Ref [10] does not consider processing of Mathematical Equations. Ref [24] makes use of Convolutional and Recurrent Neural Networks for Hand-written text analysis, although there is no feature included for mathematical equation processing.

III. METHODOLOGY

As per system architecture shown in Fig. 1, the system is split up into three major modules:

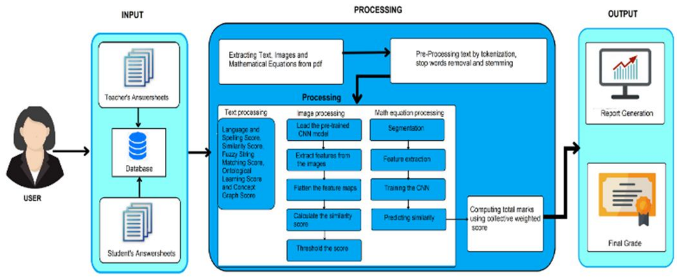


Figure 1. System Architecture Diagram

A. Pre-Processing

Pre-processing involves steps like text, diagrams, mathematical formulae extraction.

1) Extraction:

• Text Extraction

Text Extraction for student’s and teacher’s descriptive answers is done using PyPDF2 library which is effective in handling and viewing a PDF. The extracted text is then stored in the database.

• Image Extraction

Fitz module and Image module are some used for image extraction. Fitz module helps in reproducing the authentic forms of the images or diagrams. It makes use of Pixmap, which allows one to generate a raster image of a page and also contains an RGB image of the page which can therefore be used for controlling images, making grayscales, etc. Image module is used for opening, resizing and saving images. Functionalities such as filter, cropping and rotating of images are provided by Image. It also assists in modifying and extracting image data.

• Mathematical Equation Extraction

OCR software helps identify text within an image and converts it into a modifiable format. A few Optical Character Recognition (OCR) softwares can also accept mathematical formulas and transform them into LaTeX format. One can effortlessly edit and use formulae as needed once the latex code is available. Mathematical formulae can also be simply seen by a few Math OCR tools such as Mathpix OCR, InftyReader, and MathType. These tools can also turn formulae into Latex or MathML format.

2) Text Pre-processing Techniques:

• Tokenization

Tokenization helps split sentences into words, thereby, converting large texts into smaller tokens. • Stop words removal : Stop words in a text are nothing but common words that appear often. For example, stop words include ”a”, ”an”, ”the”, ”by”, ”if”, etc. This technique helps remove these stop words by eliminating redundancy.

• Lemmatization

Lemmatization is implemented to cut words to it’s base form, also known as ”root” form. For example, the word ”children” is cut down to ”child” and the word ”understood” is cut down to it’s root form, which is ”understand”.

B. Processing and Training the model

Processing includes text, image and mathematical formulae processing.

1) Text processing:

• Spell and Grammar Check Module

A python package known as SpellChecker library is used in this system that helps analyze, validate and correct spellings in a given text. It does so by providing a set of inbuilt functionalities and makes use of algorithms such as Levenshtein distance, Jaro-Winkler distance and phonetic algorithms to fix wrong spellings. Another important library put into use is the language tool python wrapper for grammar and style checking. Developers are provided with the functionality of using this module in various python applications to ensure valid usage of grammar and style in different languages.

• Similarity Score Module

If we take two non-zero vectors of an inner product space, we use cosine similarity metric to measure the similarity between these two vectors. When it comes to NLP, this specific metric is generally used to estimate the similarity between two different texts or documents.

$$\cos(\theta) = \frac{P \cdot Q}{\|P\|\|Q\|}. \qquad (1)$$

Where, P and Q are vectors that represent two text documents.

• Ontological Learning Module

Transformers are used in Ontological analysis. It makes use of a deep neural network that is trained beforehand and has the ability to encode any text into a vector representation. This is done in high dimension. The meaning of text is produced and this is used to evaluate resemblance between two texts. To elaborate, sentence transformer obtains embeddings for every sentence and then calculates the similarity.

• Fuzzy String Matching Module

To identify strings that can be identical to a target string, we use fuzzy string matching technique. This can be used for strings that are not an exact match but hold the same meaning. For this module we have further put WordNet to good use. It supplies a large collection of words and their 3 correspondence with one another. It is used for a multiple NLP tasks such as text classification, machine translation, sentiment analysis and Word Sense Disambiguation.

• Concept Graph Module

A directed graph includes directed edges and nodes that join other nodes. For drawing such easy and adaptable directed concept graphs in Python, we have used NetworkX specifically, for concept graph. In a concept graph, each node signifies a concept and the directed edge signifies some correlation between two concepts.

2) Image processing:

For image processing, there are specific steps required to assess student's hand-drawn diagrams.

• Initially, the system uses a pre-trained model like VGG16 or ResNet50 or as an alternative method, the system can be trained on a huge dataset of diagrams.

• Secondly, the pre-processed images advance their way in the CNN model and from the final convolutional layer feature maps are procured.

• Once the system procures feature maps, it is normalized between 0 or 1 by revamping it to a 1D vector.

• These normalized vectors of expected image and student's image are compared using a similarity metric like SSIM wherein, the higher the similarity goes, the greater score is generated by the system.

$$SSIM(m,n) = \frac{(2\mu_m \mu_n + C_1)(2\sigma_{mn} + C_2)}{(\mu_m^2 + \mu_n^2 + C_1)(\sigma_m^2 + \sigma_n^2 + C_2)}. \qquad (2)$$

• The bare minimum systemic similarity is signified by a regulated threshold value. If the similarity score generated by the model surpasses this threshold, the two diagrams are judged as similar.

3) Mathematical formulae processing:

For mathematical equation processing, we have used CNN in the following way:

• Preprocessing is an inevitable step to train models and the same is implemented here at the initial stage. After preprocessing, the symbols in the equation are segmented using contour detection.

• From the same segmented symbols, the system extracts features from it. Various strategies such as Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP) are applied.

• For training the CNN model, the features are put into the model and to be able to accurately classify symbols, backpropagation by modifying weights of the network is used.

• When the model is confidently trained, it predicts the amount of resemblance between two mathematical equations.

C. Post-Processing

The output generated by the system is considered here. All modules, libraries and methods implemented in pre-processing and training of system are held as parameters in calculation of the final score. Weightage will be given to individual modules based on the type of question. The question type will be considered from Bloom's Taxonomy which is a fundamental part in answer evaluation. A detailed feedback report will be generated as the output which includes Learning Outcome analysis graph. Furthermore, the feedback report will highlight grammatical and spelling errors done by the student if any and also areas or domains where the student lacks technically. Table I contains various notations used for final score calculation.

TABLE I. NOTATION FOR TOTAL SCORE CALCULATION

FS	Final Score
W_{SG}	Weight for spelling and grammar score
X_{SG}	Spelling and grammar score
W_{COS}	Weight for cosine similarity score
X_{COS}	Cosine similarity score
W_{FS}	Weight for fuzzy-string matching score
X_{FS}	Fuzzy-string matching score
W_{TS}	Weight for sentence transformer score
X_{TS}	Sentence transformer score
I_{TH}	Threshold value for diagram similarity score for diagram acceptance
X_{IMG}	Similarity score of diagrams
F_{TH}	Threshold value for maths formula similarity score for formula acceptance
X_{MF}	Similarity score of mathematical formulae
Q_{MAX}	Maximum marks of particular question
FS_1, FS_2, FS_3	Final scores of textual, diagrammatic and mathematical formulae answers respectively

$$FS_1 = \frac{(W_{SG}+W_{COS}*X_{COS}+W_{FS}*X_{FS}+W_{TS}*X_{TS})}{2} * Q_{MAX} . \quad (3)$$

$$FS_2 = \begin{Bmatrix} X_{IMG} * Q_{MAX} & if X_{IMG} \geq I_{TH} \\ 0 & otherwise \end{Bmatrix} . \quad (4)$$

$$FS_3 = \begin{Bmatrix} X_{MF} * Q_{MAX} & if X_{MF} \geq F_{TH} \\ 0 & otherwise \end{Bmatrix} . \quad (5)$$

$$FS = FS_1 + FS_2 + FS_3 . \quad (6)$$

Here, FS is the final score which is the combined score of descriptive, diagrammatic and mathematical formulae evaluation. This will be displayed in the detailed feedback report given to the student.

IV. ALGORITHMS

A. Algorithm for descriptive answer evaluation

• Start.
• Provide student's scanned answer sheet and teacher's scanned model answers.
• Preprocess the scanned answers by extracting text, tokenization, POS tagging, lemmatization.
• Find spelling and grammatical errors from the student's answers using libraries like spellchecker and language tool python.
• Find similarity score between student's answers and teacher's answers using algorithms like Cosine Similarity, Jaccard Similarity and Lexical Similarity.
• Find Fuzzy String Matching Score between student's and teacher's answers using WordNet Lemmatizer.
• Find ontological learning score and semantic similarity between the student's answers and teacher's answers using transformers like BERT, GPT, sentence transformers.
• Draw and compare concept graphs of the student's answers and teacher's answers using Networkx package, sklearn library and matplotlib library.
• Find total marks obtained by calculating summation of weighted scores from steps 4,5,6,7.
• Stop.

B. Algorithm for hand-drawn diagrams evaluation :

• Start.

• Provide student’s scanned hand-drawn diagrams and teacher’s scanned model hand-drawn diagrams.
• Resize both the diagrams to same size, convert the images to greyscale and normalize the image pixel values.
• Load pre-trained CNN models such as ResNet50, VGG16, InceptionV3, DenseNet, etc to train the model.
• Extract feature maps from the diagrams using last convolutional layer.
• Normalize the feature maps between 0 and 1.
• Calculate the structural similarity score between the two diagram using similarity metrics like cosine similarity, SSIM, Euclidean distance, etc.
• Specify a threshold value to determine a minimum acceptance score of similarity metrics.
• Stop.
C. Algorithm for hand-written mathematical formulae evaluation :
• Start.
• Provide student’s and teacher’s scanned hand-written maths formulas.
• Perform pre-processing by converting images to greyscale and resize images to same size.
• Segment each symbol in the maths formula using methods like connected component analysis, contour detection, etc.
• Perform feature extraction using methods like HOG, LBP, SIFT, etc.
• Train CNN model and adjust its weights using backpropagation.
• Calculate the structural similarity score between the two formulas using similarity metrics like cosine similarity, SSIM, Euclidean distance, etc.
• Stop.

V. FIGURES AND TABLES

Fig. 2 and Fig. 3 shows the similarity score between the two images as calculated by the system which is 0.73 0.2 respectively.

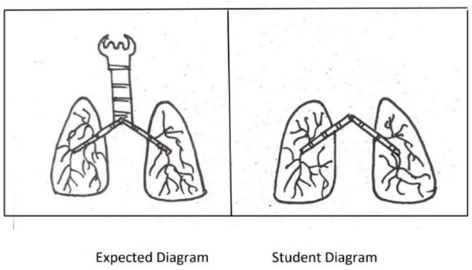


Figure 2. Comparability between Expected and Student diagram

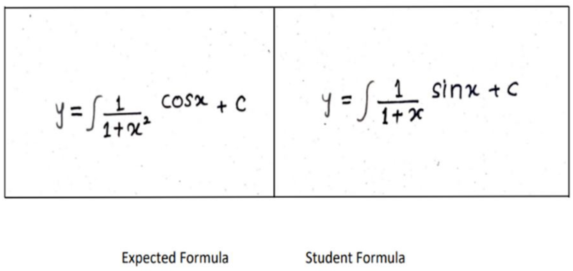


Figure 3. Comparability between Expected and Student Formula

Fig. 4 shows us the concept map generated for table II is used to inspect idea/concept similarity between the teacher's and student's answer.

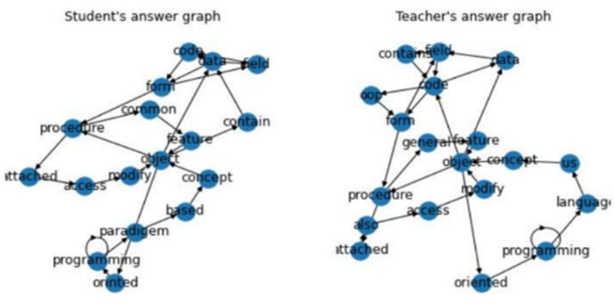


Figure 4. Comparability between teacher’s and student’s concept map

Table II contains a sample student and teacher's answer. The student's answer is not perfect and to measure it's similarity to the teacher's answer, multiple algorithms have been put to use. Table III gives a comparison for the accuracies of various used algorithms.
Table IV and table V gives a comparison for the accuracies of diagrammatical answer and mathematical answer evaluation algorithms respectively.

TABLE II. DESCRIPTIVE ANSWERS TABLE

Question	What is Object Oriented Programming
Student Answer	Object orinted programming is the standard that is on the principle of objects that have code and data. Data is in fields and code is in procedure. A comon thing of object is that that procedure are attached to them and can manipulate the object data fields.
Teacher Answer	A computer language called object-oriented programming makes use of the idea of objects. OOP uses both fields and procedures as a means of encoding its code. The fact that items have procedures linked to them is a common characteristic. The data fields of the objects are also accessible and editable. Both code and data are present.

TABLE III. DESCRIPTIVE ANSWERS EVALUATION ACCURACY

Spell Error	Grammar Error	Similarity Score	Ontology Score	Fuzzy String Score
2	1	0.84	0.95	1

TABLE IV. DIAGRAMMATICAL ANSWERS EVALUATION ACCURACY

Support Vector Machine	Convolutional Neural Network	Recurrent Neural Network	Semantic Neural Network	Structural Similarity Index	Dynamic Time Warping	Eijkhout-Wiliams-Doerfler
94.2	97.3	94.3	96.5	95.4	89.8	90.5

TABLE V. MATHEMATICAL FORMULAE ANSWERS EVALUATION ACCURACY

Support Vector Machine	Scale-Invariant Feature Transform	Convolutional Neural Network	Recurrent Neural Network	Histogram of Oriented Gradients	Self-Organizing Semantic Network	Optical Character Recognition
94.2	74.4	97.3	94.2	76.6	96.5	92.8

VI. RESULT AND DISCUSSIONS

After combining all features and attributes and grouping together descriptive, diagrammatic and mathematical answer evaluation models into a whole system, we achieved our proposed system. We implemented various state-of-the-art algorithms and compared their accuracies. This helped us select the most optimal algorithm for our model. For descriptive answer evaluation, we used various algorithms as shown in Fig. 5 and transformers like BERT, GPT and sentence transformers used for Ontological learning, gave us the highest precision of 97.6% to assess answers.

Furthermore, when it comes to Diagrammatical Answer Assessment, out of all the algorithms implemented in Fig. 6, CNN excelled in the accuracy parameter with an accuracy of 97.3%.

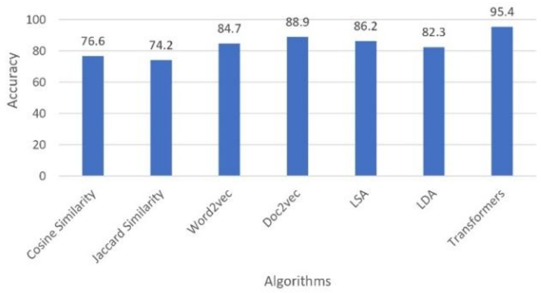


Figure 5. Algorithmic Accuracy for Descriptive Answers

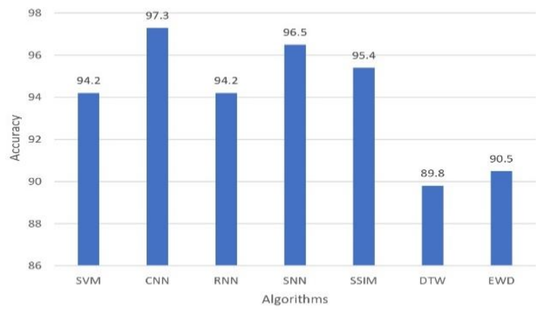
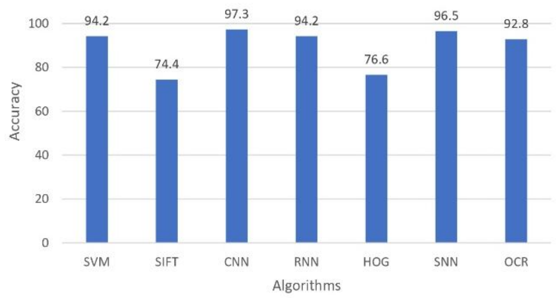


Figure 6. Algorithmic Accuracy for Diagrammatic Answers

Finally, for Mathematical Formulae Evaluation, as per Fig. 7, CNN again gave us the highest accuracy of 97.3%. A noteworthy point to observe is that SSIM algorithm also gives a tough competition to CNN by giving us an accuracy of 95.4% and 96.2% for diagrammatical and mathematical formulae evaluation respectively. To check the accuracy of the system, we took into consideration five sample questions and provided student's answers and teacher's model answers as input to the system. At the same time, the student's answers were evaluated manually. The scores generated by Manual Evaluation and System Evaluation are compared in Fig. 8.

Once the system processes the input, it works on assessing student's answers and generates a detailed feedback report based on the evaluation. The report will include a learning outcome analysis as shown in Fig. 9. These models, while working in combined harmony give us an accuracy of 97.4%.



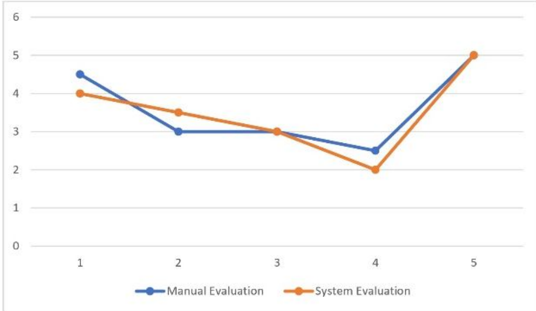


Figure 7. Algorithmic Accuracy for Mathematical Formulae

Figure 8. Relation between Manual and System Evaluation

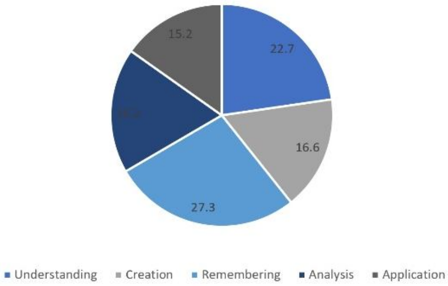


Figure 9. Learning Outcome Analysis

VII. CONCLUSION

As per the proposed system, we have implemented a student centric answer scoring tool that enhances the capabilities of the system compared to the previous systems. As seen in preceding models, similarity between two texts was mostly carried out with low precision and less optimal algorithms, unlike this system which uses better algorithms with increased optimality. In contrast to Ref [1] the proposed model provides partial scoring on non-binary basis. When all papers are studied distinctly it lacks the amalgamation of a single merged system that can assess diagrammatic and mathematical equations along with descriptive answers. Automated Descriptive and Diagrammatic Answer Scoring Tool achieves this merged system. Experimentation has revealed that this system gives an accuracy of 97.4%. This improves a major flaw in the model described in Ref [6] where the accuracy of the model was only 88%. As a consequence this proposed model pulls of high precision and algorithms at their peak performance. Although this model can be improved by adding an additional feature of speech recognition, through which viva examinations can be made easier and hassle free.

REFERENCES

[1] I. Das, B. Sharma, S. S. Rautaray and M. Pandey, “An Examination System Automation Using Natural Language Processing,” 2019 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2019, pp. 1064-1069, doi: 10.1109/ICCES45898.2019.9002048.J. Clerk Maxwell, *A Treatise on Electricity and Magnetism*, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Ramamurthy, Madhumitha & Krishnamurthi, Ilango. (2016). Design and Development of a Framework for an Automatic Answer Evaluation System Based on Similarity Measures. Journal of Intelligent Systems. 26. 10.1515/jisys-2015-0031.

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

[4] Md. Motiur Rahman, Ferdusee Akter, ”An Automated Approach for Answer Script Evaluation Using Natural Language Processing”, IJCSET(www.ijcset.net) — 2019 — Volume 9, 39-47.

[5] Sarika Singh, Yash Shah, Yug Vajani,Prof. Surekha Dholay, ”Automated Paper Evaluation System for Subjective Handwritten Answers”, 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2021.

[6] Muhammad Farrukh Bashir, Hamza Arshad, Abdul Rehman Javed, Natalia Kryvinska, Shahab S. Band, ”Subjective Answers Evaluation Using Machine Learning and Natural Language Processing, IEEE Access (Volume: 9), 2021

[7] Jagadamba G , Chaya Shree G, "Online Subjective answer verifying system Using Artificial Intelligence", Proceedings of the Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC).
[8] Dmytro Zheleznaikov, Viktor Zaytsev, Olga Radyvonenko, “Online Handwritten Mathematical Expression Recognition and Applications: A Survey”, March 2 2021, IEEE Access.
[9] Ganga Sanuvala, Syeda Sameen Fatima, “A study of Automated Evaluation of Student’s Examination Paper using Machine Learning Techniques”, 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS).
[10] Jamshed Memon, Maira Sami, Rizwan Ahmed Khan, Mueen Uddin, "Handwritten Optical Character Recognition (OCR) : A Comprehensive Systematic Literature Review (SLR)", July 16 2020, IEEE Access.
[11] Vijay Rowtula, Subba Reddy Oota, C.V. Jawahar, "Towards Automated Evaluation of Handwritten Assessments", 2019 International Conference on Document Analysis and Recognition (ICDAR).
[12] Khanh-Ngoc Bui, Quoc-Kim-Hoang Nguyen, Thanh-Sach Le, "Handwritten Mathematical Expression Recognition: An approach on data augmentation", 2021 15th International Conference on Advanced Computing and Applications (ACOMP).
[13] Mohd Azlan Abu, Nurul Hazirah Indra, Abdul Halim Abd Rahman, Nor Amalia Sapiee, Izanoordina Ahmad, "A study on Image Classification based on Deep Learning and Tensorflow", International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 4 (2019), pp. 563-569.
[14] Shaun Oommen Alexander, Chitresh Kansal, Devang Mehrotra, Rohit P, Shashank Mouli Satapathy, "A Novel Computer Vision Model for Recognition and Evaluation of Mathematical Equations using Deep Learning Technique", 2022 VLSI Device Circuit and System (VLSI DCS), IEEE.
[15] Mingle Zhou, Ming Cai, Gang Li, Min Li, "An End-to-End Formula Recognition Method Integrated Attention Mechanism", Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China.
[16] Sagar Shinde, Akil Mulagirisamy, Daulappa Bhalke, Lalitkumar Wadhwa4, "Recognition of Math Expressions & Symbols using Machine Learning", Turkish Online Journal of Qualitative Inquiry (TOJQI) Volume 12, Issue 8, July, 2021: 5987- 6000.
[17] Miss Ashlesha S. Phalke, Dr.G.R.Bamnote, Prof. Ms. S.W. Ahmad, "Evaluating Descriptive answer assessment system with Machine Learning", IJCRT.
[18] B M Vinjit, Mohit Kumar Bhojak, Sujit Kumar, Gitanjali Chalak, "A Review on Handwritten Character Recognition Methods and Techniques", International Conference on Communication and Signal Processing, July 28 - 30, 2020, India.
[19] Meena K, Lawrance Raj, "Evaluation of the Descriptive type answers using Hyperspace Analog to Language and Self-organizing Map", IEEE International Conference on Computational Intelligence and Computing Research.
[20] Md Gulzar Hussain, Sumaiya Kabir, Tamim Al Mahmud, Ayesha Khatun, Md Jahidul Islam, "Assessment of Bangla Descriptive Answer Script Digitally", International Conference on Bangla Speech and Language Processing(ICBSLP), 27-28 September, 2019.
[21] Amarjeet Kaur, M Sasikumar, "A comparative analysis of various approaches for Automated Assessment of descriptive answers", 2017 International Conference on Computational Intelligence in Data Science(ICCIDS).
[22] Rosy Salomi Victoria D, Viola Grace Vinitha P, Sathya R, "Intelligent Short Answer Assessment using Machine Learning", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958 (Online), Volume-9 Issue-4, April 2020.
[23] Eman Shaikh, Iman Mohiuddin, Ayisha Manzoor, Ghazanfar Latif *, Nazeeruddin Mohammad, "Automated Grading for Handwritten Answer Sheets using Convolutional Neural Networks", International Conference on New Trends in Computing Sciences (ICTCS).
[24] G.R. Hemanth, M. Jayasree, S. Keerthi Venii, P. Akshaya, and R. Saranya, "CNN-RNN BASED HANDWRITTEN TEXT RECOGNITION", ICTACT JOURNAL ON SOFT COMPUTING, OCTOBER 2021, VOLUME: 12, ISSUE: 01.
[25] Sridevi V, Suresh Kumar S, Supraja B, Udhayakumar S, "Knowledge Representation and Answer Evaluation System using Language Processing Algorithm", International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN).