A Hybrid Qualitative and Quantitative approach for Automatic Short Answer Grading using Classification Algorithms

Sree Lakshmi P
School of Computer Science and Applications
REVA University , Bangalore, Karnataka, India
p.sreelakshmi@reva.edu.in

Simha J B
RACE
REVA University Bangalore, Karnataka, India
jb.simha@reva.edu.in

2022 4th International Conference on Circuits, Control, Communication and Computing (I4C) | 979-8-3503-9747-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/I4C57141.2022.10057906

Abstract— **Assessment plays a very important role in the teaching-learning process. Recently proposed techniques for short answer grading are either Qualitative or Quantitative. Qualitative methods use classification and give scores in an incomprehensible way like correct, incorrect, partially correct, etc., and Quantitative methods use Regression and MSE(Mean Square Error) which assigns scores like 2.3,4.2, etc. requires a learning method to convert those values to integers. In this paper, a hybrid Qualitative Quantitative approach is proposed which is based on machine learning that relies on Multi-class classification where each category of marks is considered as a class. This can be easily converted to qualitative or quantitative systems without additional learning methods. This system follows a model answer-based method where various similarities are extracted from both model and student answers including statistical, Bag of Words, TFIDF, Latent Semantic Analysis, and Neural sentence embedding based Infersent. Using these features various classification models are designed including K Nearest Neighbor, Naïve Bayes, Decision tree, SVM, Random Forest, and XGBoost for various test sizes of the data and cross-validations. The models are tested on a dataset that consists of one question, 80 answers, and a model answer. In total, 72 experiments are conducted, demonstrating that the proposed hybrid approach can be effectively applied to the Automatic Short Answer Grading (ASAG) task.**

Keywords— ***Automatic Short Answer Grading, Automatic Evaluation, Machine Learning, Text similarity.***

I. INTRODUCTION

In teaching-learning, assessment is a significant component as it helps to identify the student's weaknesses, which further helps find the areas where improvement is required and finally realize the teaching goals more efficiently. Authentic answer grading is vital in education. But consistent and fair assessment remains a challenge.

Based on the way of assigning scores existing methods regarding short answer grading are classified into two types. The first category is Qualitative methods that use classification and produces scores in an incomprehensible way like correct or incorrect [1] [2] correct, incorrect, partially correct [3], correct, partially correct, partially incorrect, incorrect, and contradictory [4]. These methods just tell whether an answer is correct or incorrect but will not give marks. The second category is Quantitative methods [5]. which use regression and MSE that assigns scores like 2.3,4.2 etc. These methods need a learning algorithm like the nearest integer for converting the real scores to integer values. The limitation of Quantitative methods is that they can just give the marks but cannot tell us whether the answer is incorrect or correct [6], [7].

To overcome the above limitations a hybrid Qualitative and Quantitative approach for data interpretation based on machine learning is proposed. Here Automatic Short answer Grading (ASAG) task is treated as a Multi-class classification problem that considers each mark as a label. For example, regarding a 5 marks question, the labels are 0,1,2,3,4,5. Because one cannot be so subjective to assign marks like 3.4 or 2.3 for 5 marks. This model can be easily converted as qualitative just by using a filter without any additional learning algorithm. The proposed system is also quantitative as it behaves like an actual human interpreter giving marks 1,2,3,4 etc. and it mimics more human cognition and action than a machine-oriented system. The work inspected various classifiers and methods to examine which combination of learners and features produces reliable outputs.

The core component of the proposed model is the model answer-based approach. For comparison, similarities between the model and student answers are extracted using techniques like word occurrence statistics, TFIDF, Latent semantic analysis, Semantic similarity using Infersent, and summary similarity using extractive summarization. These extracted similarities are given as input features to various classification algorithms like KNN, Naïve Bayes, SVM, Random Forest, and XGBoost, and models are designed for samples including train test split and cross-validation. Overall, 72 experiments are explored for various samples and classifiers. The model is tested using a dataset that consists of 80 answers for a Question and a model answer whose scale is 0-4 and evaluated using metrics including Accuracy, AUC, Precision, Recall, and F score. Experimental results prove that the decision tree classifier with Gini Criterion outperforms the remaining classifiers with the best accuracy.

This paper is designed as follows: Section 2 describes the related work, Section 3 presents the proposed system that has been anticipated to evaluate descriptive answers, Section 4 is designated for the results and discussions and finally, Section 5 presents the conclusion and future scope.

979-8-3503-9747-5/22/$31.00 ©2022 IEEE

II. LITERATURE SURVEY

Automatic short answer grading is a predominant research area early from the 1960s. Several systems have been designed for essay and short answer grading [8] The concept mapping method considers student answers as a collection of concepts and tries to perceive the absence or presence of each concept while grading. For example (ATM) Automatic Text Marker [9] ,C-rater(Concept rater)[10].The information extraction methods mainly focus on finding the facts in student answers. Here, pattern-matching operations like parse trees, Regular expressions, etc. were considered for grading. For Example, Auto mark [11], Emax [12] and Indus Marker [14].Next, Corpus-based methods mainly focus on the statistical properties of huge documents. Example Atenea [15].

The machine learning systems utilize measurements extracted from NLP techniques and find the grades using classification or regression models. For example, CAM, Content Assessment module [16] Whereas[17] extracted features like POS tags, TF-IDF, term frequency, and entropy along with the SVM classifier. [18] proposed a semiautomated approach wherein they used statistical corpus-based features like tf-idf, length, LSA similarity based on Wikipedia, etc, and classical NLP features.[19] used WordNet graphs to find the text similarity between the Model and student answers.[20] proposed stacking model based on XGBoost and Neural network and data upsampling method to address the data imbalance problem.[21] proposed a method to leverage the Question and student models to a reference answer-based approach and explored the effectiveness of the Deep Belief Network (DBN) for the ASAG task.

III. PROPOSED METHOD

Here, the proposed methodology for automatic short-answer grading is described . The roadmap of the approach is given in Fig 1 and the specific components discourse below.

A. Data Collection

A real-world dataset is collected that consists of one question and 80 student answers and one model answer with a score of 4 marks. These student answers are evaluated by the teacher and grades are given in the range of 0 to 4. Here 0 represents that the answer is incorrect and 4 represents the answer is correct whereas grades 1,2,3 denote a partially correct answer. Table 1 shows the sample question, Model answer, and score. Table 2 shows a snippet of student answers and their corresponding scores for the given question.

B. Data Preprocessing

Data pre-processing techniques mentioned in Table 3 are applied to the data to make it ready for model building. For this Natural Language Toolkit or NLTK python library is used for the task of data pre-processing. Table 4 below shows a snippet of Student answers after performing Data Pre-processing [17], [22], [23].

TABLE 1. THE QUESTION, MODEL ANSWER, AND SCORE

Question	Model answer	Score
Define Cloud computing with an example.	Cloud Computing is a model for enabling convenient and on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) which can be quickly provisioned and released through minimal service provider interaction or management effort. For example, Google hosts a cloud that consists of both larger servers and small PCs. It is a private one (that is, owned by Google) which is publicly accessible (by Google's users).	**4**

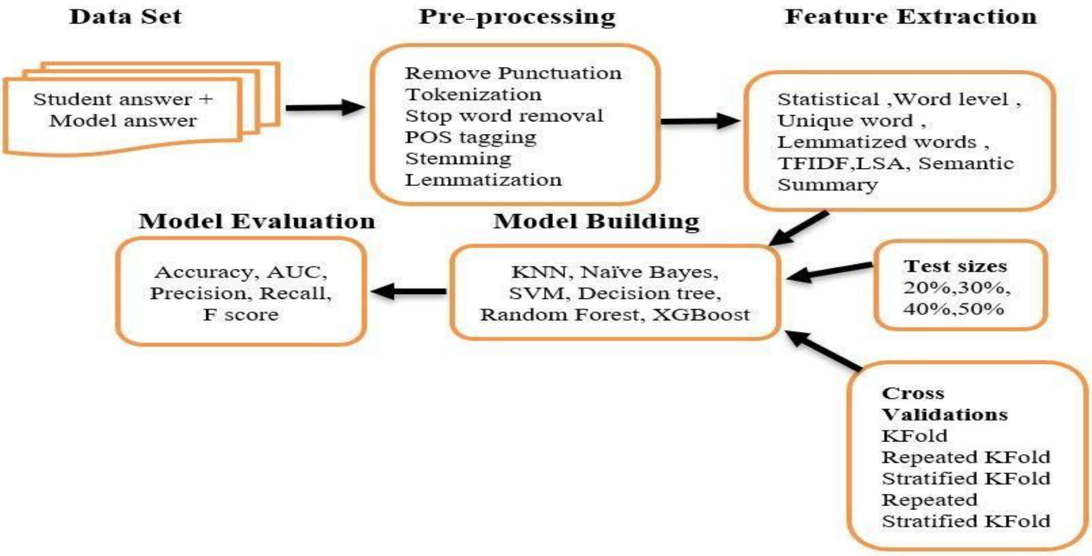
Data Set
Student answer + Model answer
Pre-processing
Remove Punctuation
Tokenization
Stop word removal
POS tagging
Stemming
Lemmatization
Feature Extraction
Statistical ,Word level , Unique word , Lemmatized words , TFIDF,LSA, Semantic Summary
Model Evaluation
Accuracy, AUC, Precision, Recall, F score
Model Building
KNN, Naïve Bayes, SVM, Decision tree, Random Forest, XGBoost
Test sizes
20%,30%, 40%,50%
Cross Validations
KFold
Repeated KFold
Stratified KFold
Repeated Stratified KFold

Fig1:Proposed Methodology for Automatic Short Answer Grading

TABLE 2. STUDENT ANSWERS AND THEIR CORRESPONDING SCORES GIVEN BY THE TEACHER

SID	Student answer	Scores
1.	It is a model for enabling convenient network access to a shared pool of computing resources Ex google cloud	1
…	---	
80	Cloud computing model enables convenient access to network resources with minimum management effort.	1

C. Data Sampling

Here the pre-processed data is sampled using train test split and cross-validation methods. The classification models are designed by considering the 20%,30%,40%,and 50% test sizes of the data.

The models are also designed using various Cross Validation methods like KFold, Repeated KFold, Stratified KFold, and Repeated Stratified KFold.

D. Feature Extraction :

Here the following similarities are extracted from the model and student answers.

TABLE 3. DATA PREPROCESSING TECHNIQUES AND THEIR DESCRIPTION

Sl No	Preprocessing Technique	Description
1	Removing Punctuation / special characters	Removing all Unnecessary characters and special symbols
2	Tokenization	Splitting the sentences into individual words called tokens
3	Stop words Removal	Removing commonly used words like a, an, the, etc.
4	POS tagging	To understand the meaning of the text For every word Parts of Speech is determined and tagged.
5	Stemming	Here words are transformed into their root form by removing the prefix or suffix. Ex: “Studies” converted into “studi”
6	Lemmatization	Here words are converted into their base form by considering the morphological analysis. Ex: “Studies” converted into “study”.

TABLE 4. A SNIPPET OF STUDENT ANSWER AFTER DATA PREPROCESSING

Preprocessing technique	Student Answer
Student answer	It is a model for enabling convenient network access to a shared pool of computing resources. Ex: google cloud
After removing Punctuation	It is a model for enabling convenient network access to a shared pool of computing resources Ex google cloud
After Tokenization	['it', 'is', 'model', 'for', 'enabling', 'convenient', 'network', 'access', 'to', 'a', 'shared', 'pool', 'of', 'computing', 'resources', 'google', 'cloud']
After removing Stop words	['model', 'enabling', 'convenient', 'network', 'access', 'shared', 'pool', 'computing', 'resources', 'google', 'cloud']
After POS tagging	[('It', 'PRP'), ('model', 'VBZ'), ('enabling', 'VBG'), ('convenient', 'NN'), ('network', 'NN'), ('access', 'NN'), ('shared', 'VBD'), ('pool', 'NN'), ('computing', 'VBG'), ('resources', 'JJ'), ('google', 'NN'), ('cloud', 'NN')]
After Stemming	['model', 'enabl', 'convini', 'network', 'access', 'share', 'pool', 'comput', 'resources', 'googl', 'cloud']
After Lemmatization	['model', 'enabling', 'convenient', 'network', 'access', 'shared', 'pool', 'computing', 'resources', 'google', 'cloud']

1) Statistical Similarity

Statistical information like the number of sentences, count of unique words, count of words, count of nouns, prepositions, verbs, discourse connectives, and adjectives are extracted from the model and student answers. Each answer is represented as a vector with statistical values as dimensions using the vector similarity approach.

2) Word-word Similarity

It is based on how many words are matching in both the model and student answers. To achieve this both model and student answers are represented as a Bag of Words using Count Vectorizer and tried to find the cosine similarity between the two bags of words [5], [13].Then the Euclidean distance between the vectors of the model and student answers is considered for similarity. If the similarity is high, then the answer gets a high score [5], [6], [24].

3) Nonstop word similarity /Unique word similarity

In Word-Word Similarity, stop words are considered while calculating the similarity. But in most of the situations stop words may not play a significant role, So the similarity without stop words also calculated. For this, a bag of words are generated for both model and student answers using the Count vectorizer with parameter stop_words = ‘english’.This will extract unique words by removing the stop words. Then cosine similarity between the model and student unique words is calculated [5].

4) Lemmatized word Similarity

Here the similarity between the root words of the student and model answers is calculated. For this, the lemmatized bag of words of model and student answers are generated using Count Vectorizer. Cosine similarity is measured between these two lemmatized vectors.

5) TFIDF similarity

The disadvantage of Bag of words vectors is the large vocabulary size as it considers stop words also and no information regarding the grammar of sentences or ordering of words in the text will be retained. To overcome the above disadvantages TFIDF vectors are created for both model and student answers and cosine similarity between those vectors is calculated [6], [18], [24].

6) LSA similarity

Either BoW or TFIDF vectors will not consider the context of a sentence. The contextual meaning of words is extracted and represented using Latent Semantic Analysis (LSA). Here, first represented the text as TFIDF vectors, then Singular Value decomposition SVD is applied to perform dimensionality reduction. The similarity is calculated by the dot product of those reduced vectors [5].

7) Semantic Similarity

The Semantic similarity between the model and student answer is calculated using Infersent [25]. It is an NLP technique for sentence embeddings based on the Bi LSTM with mean-max pooling for creating a vector of fixed size. . The sentence embeddings of both model and student answers are built using Infersent and cosine similarity are computed [5].

8) Summary similarity

Here, the similarity between the model and student answers at the summary level is calculated . First using extractive summarization the summaries of both model and reference answers are created then created vectors of both summaries and calculated cosine similarity as a similarity measure [5].

E. Model Building

Here the following 9 classification models are designed by considering the above 8 similarity features as independent and Scores of the answers as a dependent feature. The models are built using the modules from sci-kit learn and below is the explanation of each classification model.

The KNN Classifier-without scaled features(KNN-NS) is designed by considering the features without any scaling. In the case of Non Scaled features, the optimal K value was 7 using the elbow method [23]. Second, KNN Classifier with scaled features (KNN-S) is designed to test how scaling (standard scalar) affects the model's performance. With scaled features, the optimal K value is 1.

The third is a Decision tree classifier with Information Gain (DT_IG) criteria. Fourth is the decision tree classifier with Gini Criteria(DT-GINI). The fifth classifier is the Naïve Bayes classifier (NB) using Gaussian Naïve Bayes wherein simple Gaussian distribution is used to draw the data from each label [21] Sixth is the SVM classifier(SVM) used Scikit-learn SVC support vector machine classifier by initializing with its default values. Here the values of the features are scaled using a standard scalar to represent all values in a particular range with mean zero and standard deviation as one [17], [21], [26]. Seventh is the Random Forest Classifier (RF) by considering the default parameters and all computational linguistic features as input [26], [27]. Eighth is the Random Forest Classifier with tuned Hyperparameters (RF-HT). The best hyperparameters for the Random Forest classifier are selected using Randomized Search CV and Grid Search CV. Random search permits us to narrow down the range for each hyperparameter and with the Grid search CV the different combinations of settings can be specified explicitly [28]. Finally, the ninth is the XGBoost Classifier using the default parameters of XGBoost [20]**.**

F. Model Evaluation Module

Once classification models are designed, it is very important to evaluate them to know the efficiency of the model. The following five evaluation metrics are used.

i) *Accuracy:* It is the most widely used metric and indicates how many answers are graded correctly.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (1)$$

ii) Area under the curve(AUC): It is the area under the receiver operating characteristic(ROC) curve which is created by plotting the true-positive rate(TPR) against the false-positive rate(FPR) at different threshold settings. In the case of an imbalanced dataset, AUC is preferable to accuracy.

iii) Precision: It specifies out of all predicted positive results how many are actually positive.

$$Precision = \frac{TP}{TP+FP} \qquad (2)$$

iv) Recall: It specifies out of all total positive classes how many positive classes were predicted correctly.

$$Recall = \frac{TP}{TP+FN} \qquad (3)$$

Note: Here TP = True Positives, TN = True Negatives, FP= False Positives, FN = False Negatives

v) F Score: It provides an effective assessment by considering both precision and Recall

$$F - Score = \frac{2*Precision*Recall}{Precision+Recall} \qquad (4)$$

III. RESULTS AND DISCUSSION

Here, the results achieved from the experiments conducted on the dataset that consists of one question, a model answer, and sample answers from 80 students are presented. The maximum mark for the question is 4. In the proposed method the numeric scores 0 to 4 are assigned for the answers and each score is considered as an individual class label. Here a score of 0 indicates that the answer is completely incorrect or irrelevant and a score of 4 indicates perfect answers. Nine machine learning classifiers are designed for various test sizes 20%,30%, 40%, and 50% cross-validation methods KF(KFold,10 Fold), RKF (Repeated KFold), SKF (Stratified KFold), and RSKF (Repeated Stratified K-fold. Experimental results show that DT-GINI for a test size of 30%, outperforms the remaining classifiers. The performance of classifiers in terms of accuracy, AUC score, Precision, Recall, and F score for various test sizes and cross-validations can be seen in Table 5, Table 6, Table 7, Table 8, and Table 9.

TABLE 5. ACCURACY VALUES OF NINE CLASSIFIERS FOR VARIOUS TEST SIZES AND CROSS-VALIDATIONS

Model	**20%**	**30%**	**40%**	**50%**	**KF**	**RKF**	**SKF**	**RSKF**
KNN-NS	0.56	0.66	0.56	0.55	0.71	0.74	0.76	0.75
KNN-S	0.37	0.58	0.62	0.62	0.66	0.66	0.66	0.66
DT - IG	0.62	0.79	0.71	0.72	0.68	0.67	0.66	0.69
DT-GINI	0.68	0.79	0.62	0.72	0.76	0.72	0.78	0.75
NB	0.62	0.70	0.53	0.6	0.65	0.62	0.62	0.62
SVM	**0.81***	0.66	0.65	0.62	0.72	0.74	0.77	0.77
RF	0.56	0.62	0.53	0.55	0.72	0.69	0.73	0.72
RF-HT	0.56	0.66	0.56	0.6	0.73	0.7	0.76	0.74
XG	0.43	0.58	0.56	0.65	0.7	0.67	0.72	0.71

A few observations from the results are :

From Table 5 it is observed that the maximum accuracy is achieved by the SVM classifier for a test size of 20%, which

TABLE 6. AUC SCORE OF NINE CLASSIFIERS FOR VARIOUS TEST SIZES AND CROSS-VALIDATIONS

Model	20%	30%	40%	50%	KF	RKF	SKF	RSKF
KNN-NS	0.87	0.94	0.91	0.89	0.85	0.88	0.92	0.92
KNN-S	0.45	0.5	0.5	0.5	0.73	0.74	0.81	0.80
DT - IG	0.83	0.88	0.74	0.88	0.82	0.82	0.87	0.85
DT-GINI	0.89	**0.96***	0.88	0.88	0.84	0.86	0.92	0.90
NB	0.91	0.95	0.94	0.95	0.84	0.86	0.94	0.92
SVM	0.89	0.92	0.90	0.89	0.85	0.88	0.91	0.92
RF	0.86	0.95	0.92	0.92	0.86	0.88	0.95	0.93
RF-HT	0.88	0.94	0.93	0.92	0.84	0.88	0.94	0.94
XG	0.78	0.90	0.90	0.91	0.84	0.85	0.88	0.89

TABLE 7. PRECISION VALUES OF NINE CLASSIFIERS FOR VARIOUS TEST SIZES AND CROSS-VALIDATIONS

Model	20%	30%	40%	50%	KF	RKF	SKF	RSKF
KNN-NS	0.72	0.82	0.61	0.42	0.62	0.72	0.73	0.75
KNN-S	0.40	0.66	0.67	0.72	0.57	0.60	0.65	0.66
DT - IG	0.68	0.80	0.65	0.79	0.67	0.64	0.61	0.65
DT-GINI	0.80	**0.89***	0.50	0.78	0.69	0.69	0.78	0.74
NB	0.69	0.79	0.47	0.53	0.62	0.60	0.62	0.64
SVM	0.88	0.74	0.52	0.54	0.69	0.72	0.80	0.78
RF	0.72	0.77	0.57	0.59	0.67	0.68	0.73	0.74
RF-HT	0.66	0.78	0.58	0.64	0.67	0.68	0.78	0.77
XG	0.45	0.65	0.59	0.66	0.63	0.64	0.71	0.71

TABLE 8. RECALL VALUES OF NINE CLASSIFIERS FOR VARIOUS TEST SIZES AND CROSS-VALIDATIONS

Model	20%	30%	40%	50%	KF	RKF	SKF	RSKF
KNN-NS	0.65	0.77	0.68	0.49	0.63	0.73	0.77	0.77
KNN-S	0.49	0.73	0.74	0.72	0.57	0.60	0.7	0.69
DT - IG	0.67	0.83	0.61	0.78	0.63	0.64	0.69	0.69
DT-GINI	0.75	**0.86***	0.72	0.78	0.68	0.71	0.81	0.77
NB	0.72	0.81	0.66	0.68	0.61	0.61	0.66	0.66
SVM	0.84	0.77	0.74	0.69	0.68	0.73	0.79	0.78
RF	0.65	0.75	0.67	0.66	0.67	0.69	0.75	0.76
RF-HT	0.67	0.79	0.69	0.69	0.68	0.69	0.8	0.78
XG	0.54	0.73	0.69	0.73	0.63	0.63	0.75	0.73

indicates that SVM works well for small test data also. But as our data is imbalanced, only accuracy cannot be considered for identifying the best model. The DT-GINI classifier for test 30% outperforms other models in terms of AUC 96%, Precision 89%, Recall 86% and F-Score 86% which indicates it works well for imbalanced datasets.

The Naïve Bayes classifier falls behind other classifiers in almost all cases which indicates the dataset does not satisfy the prerequisite that all dependent features must be conditionally independent of each other. As the dataset is small the random forest and XGBoost did not show prominent results. The model can be tested by applying to large datasets and test their performance.

TABLE 9. F SCORE VALUES OF NINE CLASSIFIERS FOR VARIOUS TEST SIZES AND CROSS-VALIDATIONS

Model	20%	30%	40%	50%	KF	RKF	SKF	RSKF
KNN-NS	0.68	0.79	0.64	0.45	0.61	0.70	0.73	0.74
KNN-S	0.42	0.67	0.65	0.69	0.57	0.58	0.65	0.65
DT - IG	0.61	0.75	0.62	0.73	0.62	0.61	0.62	0.64
DT-GINI	0.77	**0.86***	0.55	0.73	0.68	0.68	0.77	0.73
NB	0.69	0.79	0.51	0.58	0.59	0.58	0.61	0.62
SVM	0.85	0.73	0.57	0.59	0.66	0.70	0.77	0.76
RF	0.68	0.76	0.54	0.59	0.66	0.67	0.72	0.73
RF-HT	0.66	0.78	0.56	0.61	0.67	0.67	0.76	0.76
XG	0.47	0.67	0.58	0.66	0.61	0.61	0.70	0.7

*Note: In above tables 20%, 30%, 40% and 50% represents test data percentage. KF= KFold, RKF= Repeated KFold, SKF = Stratified KFold, RSKF = Repeated Stratified KFold. The best model values under each metric are mentioned in bold and *.*

Fig 2 is the heatmap of the confusion matrix for the Decision tree classifier using the Gini criterion for a test size of 30%. The diagonal elements represent the entire correct values predicted per class. Here 19 values have been correctly predicted and 5 values are misclassified wherein 4 are from class 2 which are misclassified as class 1. The model is not able to classify class 2 accurately. The lighter color of the diagonal element(1,1) represents that our model performs well for class 1.

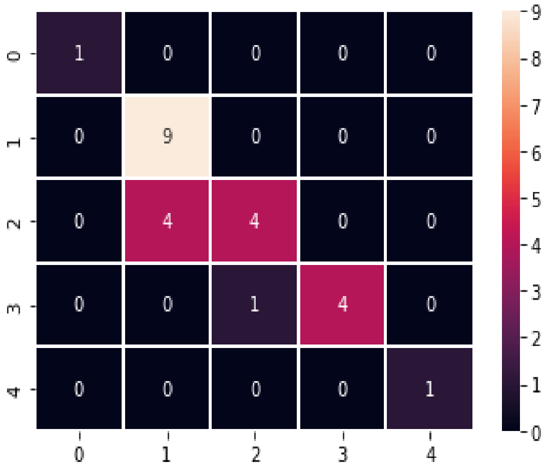

Fig 2. Confusion matrix heatmap of a Decision tree classifier

IV. CONCLUSION AND FUTURE SCOPE

In this paper, a hybrid approach of qualitative and quantitative data interpretation for the automatic grading of descriptive answers is presented. Qualitative approaches use classification and incomprehensibly provide results whereas quantitative approaches use regression and require a discriminator for converting the results into integers. Our approach overcomes the above limitations and can be used both as a qualitative and quantitative system. Explored the use of classification algorithms for Short Answer Grading by considering it as a multiclass classification problem in which each mark is treated as a label/class. The experimental results show that our hybrid approach works well for Automatic short-answer grading. The limitation of the study is, that the proposed model is applied to a small dataset of 1 Question, 1 model answer, and 80 student answers. Further, it can be evaluated on large datasets and benchmark datasets like Mohler and SemEval datasets.Some possible directions to improve the system are to build Stacking and deep learning models. Handling Questions that have multiple reference answers. Currently, sequence semantic similarity is not implemented.LSTM or Bi LSTM can be used to achieve it. Validation of marks done quantitatively using classification can be verified using regression. Can build models that can evaluate the answers with non-textual data like diagrams, tables, and equations.

Acknowledgment

The authors acknowledge the support from REVA University for the facilities provided to carry out the research.

References

[1]. R. A. Rajagede and R. P. Hastuti, "Stacking Neural Network Models for Automatic Short Answer Scoring," IOP Conf Ser Mater Sci Eng, vol. 1077, no. 1, p. 012013, Feb. 2021, doi: 10.1088/1757-899x/1077/1/012013.

[2]. A. Condor, "Exploring Automatic Short Answer Grading as a Tool to Assist in Human Rating," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2020, vol. 12164 LNAI, pp. 74–79. doi: 10.1007/978-3-030-52240-7_14.

[3]. W. J. Hou and J. H. Tsao, "Automatic assessment of students' free-text answers with different levels," International Journal on Artificial Intelligence Tools, vol. 20, no. 2, pp. 327–347, Apr. 2011, doi: 10.1142/S0218213011000188.

[4]. J. Lun, J. Zhu, Y. Tang, and M. Yang, "Multiple Data Augmentation Strategies for Improving Performance on Automatic Short Answer Scoring." [Online]. 2020, Available: www.aaai.org

[5]. S. K. Saha and R. Gupta, "Adopting computer-assisted assessment in evaluation of handwritten answer books: An experimental study," Educ Inf Technol (Dordr), vol. 25, no. 6, pp. 4845–4860, Nov. 2020, doi: 10.1007/s10639-020-10192-6.

[6]. S. Lakshmi, "Document Representation Methods for Text Categorization : A Review," International Journal of Scientific Research in Computer Science Applications and Management Studies, vol. 7, no. 6, Nov. 2018.

[7]. P. S. Lakshmi, "Intelligent Scoring Systems for Descriptive Answers-A Review,", Test Engineering and Management no. 3595, pp. 3595–3600, 2020.

[8]. S. Burrows, I. Gurevych, and B. Stein, "The eras and trends of automatic short answer grading," International Journal of Artificial Intelligence in Education, vol. 25, no. 1. Springer New York LLC, pp. 60–117, Jan. 10, 2015. doi: 10.1007/s40593-014-0026-8.

[9]. D. Callear, J. Jerrams-Smith, and V. Soh, "CAA OF SHORT NON-MCQ ANSWERS," United KIngdom, 2001.

[10]. C. Leacock and M. Chodorow, "C-rater: Automated Scoring of Short-Answer Questions," 2003.

[11]. T. Mitchell and T. Russell, "Towards robust computerised marking of free-text responses Understanding evolution and inheritance in the national curriculum KS2-3 View project GEMSTONE technology: optimisation of global supply chain View project," 2002.

[12]. D. , S. B. , S. S. , M. A. Sima, "Intelligent Short Text Assessment in eMax," 2009.

[13]. L. Cutrone, M. Chang, and Kinshuk, "Auto-assessor: Computerized assessment system for marking student's short-answers automatically," in Proceedings - IEEE International Conference on Technology for Education, T4E 2011, 2011, pp. 81–88. doi: 10.1109/T4E.2011.21.

[14]. R. Siddiqi and C. Harrison, "A systematic approach to the automated marking of short-answer questions," in IEEE INMIC 2008: 12th IEEE International Multitopic Conference - Conference Proceedings, 2008, pp. 329–332. doi: 10.1109/INMIC.2008.4777758.

[15]. E. Alfonseca and D. Pérez, "Automatic Assessment of Open Ended Questions with a BLEU-inspired Algorithm and shallow NLP," 2004. Accessed: Feb. 03, 2022. [Online]. Available: http://alfonseca.org/pubs////2004-estal2.pdf

[16]. S. Bailey and D. Meurers, "Diagnosing meaning errors in short answers to reading comprehension questions," 2008.

[17]. Wen-Juan Hou, Jia-Hao Tsao, Sheng-Yang Li, and Li Chen, "Automatic Assessment of Students' Free-Text Answers with Support Vector Machines," in Proceedings of the 23rd international conference on industrial engineering and other applications of applied intelligent systems, 2010, pp. 235–243.

[18]. Sumit Basu, Chuck Jacobs, and Lucy Vanderwende, "Powergrading: a Clustering Approach to Amplify Human Effort for Short Answer Grading," in Transactions of the Association for Computational Linguistics, Oct. 2013, pp. 391–402.

[19]. S. Vij, D. Tayal, and A. Jain, "A Machine Learning Approach for Automated Evaluation of Short Answers Using Text Similarity Based on WordNet Graphs," Wirel Pers Commun, vol. 111, no. 2, pp. 1271–1282, Mar. 2020, doi: 10.1007/s11277-019-06913-x.

[20]. R. A. Rajagede and R. P. Hastuti, "Stacking Neural Network Models for Automatic Short Answer Scoring," IOP Conf Ser Mater Sci Eng, vol. 1077, no. 1, p. 012013, Feb. 2021, doi: 10.1088/1757-899x/1077/1/012013.

[21]. Y. Zhang, C. Lin, and M. Chi, "Going deeper: Automatic short-answer grading by combining student and question models," User Model User-adapt Interact, vol. 30, no. 1, pp. 51–80, Mar. 2020, doi: 10.1007/s11257-019-09251-6.

[22]. Andrea Horbach, Alexis Palmer, and Manfred Pinkal, "Using the text to evaluate short answers for reading comprehension exercises," in Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task, Jun. 2013, pp. 286–295.

[23]. D. M. Stacey Bailey, "Diagnosing meaning errors in short answers to reading comprehension questions," in Proceedings of the Third ACL Workshop on Innovative Use of NLP for Building Educational Applications, 2008, pp. 107–114.

[24]. P. S. Lakshmi, "An Improved Hybrid Stacked Classifier For Multi Label Text Categorization," no. 3, pp. 5911–5915, 2019, doi: 10.35940/ijrte.C4739.098319.

[25]. A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised Learning of Universal Sentence Representations from Natural Language Inference Data," May 2017, [Online]. Available: http://arxiv.org/abs/1705.02364

[26]. A. Elnaka, O. Nael, H. Afifi, and N. Sharaf, "AraScore: Investigating Response-Based Arabic Short Answer Scoring," in Procedia CIRP, 2021, vol. 189, pp. 282–291. doi: 10.1016/j.procs.2021.05.091.

[27]. S. Saha, T. I. Dhamecha, S. Marvaniya, R. Sindhgatta, and B. Sengupta, "Sentence level or token level features for automatic short answer grading?: use both," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2018, vol. 10947 LNAI, pp. 503–517. doi: 10.1007/978-3-319-93843-1_37.

[28]. A. Sahu and P. K. Bhowmick, "Feature Engineering and Ensemble-Based Approach for Improving Automatic Short-Answer Grading Performance," IEEE Transactions on Learning Technologies, vol. 13, no. 1, pp. 77–90, Jan. 2020, doi: 10.1109/TLT.2019.2897997.