

An automated essay evaluation system using natural language processing and sentiment analysis

Vijaya Shetty Sadanand¹, Kadagathur Raghavendra Rao Guruvyas¹, Pranav Prashantha Patil¹,
Jeevan Janardhan Acharya¹, Sharvani Gunakimath Suryakanth²

¹Department of Computer Science and Engineering, Nitte Meenakshi Institute of Technology, Bengaluru, India

²Department of Computer Science and Engineering, RV College of Engineering, Bengaluru, India

Article Info

Article history:

Received Sep 3, 2021

Revised Jul 23, 2022

Accepted Aug 18, 2022

Keywords:

Essay scoring

Long short-term memory

Natural language processing

Plagiarism check

Sentiment analyzer

ABSTRACT

An automated essay evaluation system is a machine-based approach leveraging long short-term memory (LSTM) model to award grades to essays written in English language. natural language processing (NLP) is used to extract feature representations from the essays. The LSTM network learns from the extracted features and generates parameters for testing and validation. The main objectives of the research include proposing and training an LSTM model using a dataset of manually graded essays with scores. Sentiment analysis is performed to determine the sentiment of the essay as either positive, negative or neutral. The twitter sample dataset is used to build sentiment classifier that analyzes the sentiment based on the student's approach towards a topic. Additionally, each essay is subjected to detection of syntactical errors as well as plagiarism check to detect the novelty of the essay. The overall grade is calculated based on the quality of the essay, the number of syntactic errors, the percentage of plagiarism found and sentiment of the essay. The corrected essay is provided as feedback to the students. This essay grading model has gained an average quadratic weighted kappa (QWK) score of 0.911 with 99.4% accuracy for the sentiment analysis classifier.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Vijaya Shetty Sadanand

Department of Computer Science and Engineering, Nitte Meenakshi Institute of Technology

Bengaluru-560064, Karnataka, India

Email: vijayashetty.s@nmit.ac.in

1. INTRODUCTION

In recent years, there has been a surge in the growth of internet-related technology. Because of the advanced methods by which students learn, prepare, and undergo the examination, online exams have gained significant importance in educational institutions [1]. In today's world, not only examiners but also computers such as automatic essay evaluation are used to mark essays, since manual grading is time-consuming and prone to inconsistency and errors. In applications that perform automatic essay grading, the latent semantic analysis is used as a data reclaim tool [2]–[5]. This automated essay evaluation system grades essays on a scale of 1-10 and eliminates the time-consuming process of manually grading the essays. The system can perform anytime grading of essays unlike the manual grading process which is normally performed during the working hours of the evaluator [6], [7].

Sentiment analysis, also known as sentiment or polarity classification analyzes student's opinion from the essay for the provided topic [8]. The polarity can either be positive, negative or neutral. His system performs grading of essays based on the features extracted from the essay using natural language processing (NLP) [9]. The machine learning model leverages long short-term memory (LSTM) network to extract

features from the submitted essay. The extracted features are used to further create feature vectors to judge the quality of the essay using LSTM model. The syntactic errors are detected using the language tool which includes the grammatical and spelling mistakes performed by the student in the essay. The system also checks for the novelty of the essay through a plagiarism check using Bing search from various web sources. A naïve Bayes classifier is used to build the sentiment classifier [10].

The quality of the essay is graded on a scale of 3, the marks for syntactic errors and plagiarism percentage is deducted from a scale of 3 each. If the sentiment of the essay does not agree with the expected sentiment for the topic provided, 1 mark is deducted from the total score. Based on the syntactic errors detected, the essay is modified and the corrected essay without any errors is provided to the student and the detected errors are accompanied with the type of the error, and an example as part of the feedback to the students.

2. RESEARCH METHOD

The proposed automated essay scoring system uses NLP and sentiment analysis for prediction of grades and to comment on the student's approach towards a topic respectively. The pandas package is used to read the training dataset into a data frame. The irrelevant columns of the data frame are dropped off to ease the process of training [11]–[13].

2.1. Dataset for automatic essay grading system

The dataset used is collected from a platform related to data science called Kaggle. The dataset is provided by the Hewlett Foundation. It contains 12,000 essays written by students belonging from 7th to 10th grade. Each essay is classified into one of the 8 sets where each set contains essays written for a specific topic. Each essay is evaluated by two manual evaluators.

2.2. Dataset for sentiment analysis

A twitter dataset is gathered by utilizing the twitter API, furthermore, the information is commented as either positive or negative tweets. The dataset is openly accessible in the natural language toolkit (NLTK) corpora resource, widely investigated in numerous examinations. The complete corpus size is 10,000 tweets, comprising of 5,000 positive posts and 5,000 negative posts. We accumulated and arranged informational indexes as 7,000 tweets for training the model and 3,000 tweets for validating the model.

2.3. Building a model

The LSTM model was found suitable for this application [14]. A sequential neural network model is defined consisting of two LSTM layers, a dropout layer with a dropout rate of 0.5 and a densely connected layer using rectified linear unit (ReLU) activation function with a single output neuron. The defined model is compiled using mean squared error loss function, RMSprop optimizer and accuracy as the evaluation metric. The model is trained using K fold process for 5 folds using the training dataset. Language tool is used to detect grammar, spelling and syntactic errors in the submitted essay. Plagiarism check is performed on the essay by referring to the websites with similar content and a percentage score is provided indicating the amount of plagiarized content from each detected website.

The architecture design of the system is shown in Figure 1. It shows a high-level view of the working of the system. The teacher posts a topic and the student submits the essay to the essay evaluator which is further subjected to feature extraction, sentiment analysis and plagiarism check to generate grades accordingly.

2.4. Sentimental analysis

To detect the sentiment of the submitted essay, a naïve bayes classifier is built. The classifier is trained on the twitter dataset that contains about 10,000 tweet samples categorized as positive or negative accordingly. Randomly shuffled 7,000 tweets are used for training purpose and the remaining 3,000 for testing the classifier. Figure 2 shows the flow of sentiment analysis. It takes the twitter dataset as an input parameter and produces a sentiment classifier that is capable of classifying an essay as having either positive or negative sentiment [15], [16].

2.5. Plagiarism detection

This phase determines if the student has copied information from any web sources and lists the uniform resource locator (URL) links of the websites along with the amount of plagiarism detected from each of them and also the total percentage of plagiarism. This process consists of determining the websites with relevant content and calculating the amount of similarity between the submitted essay and the text in the

website [17]. The algorithm takes an essay as input and provides a dictionary as output with plagiarized URL as the key and the plagiarism percentage as the value [18].

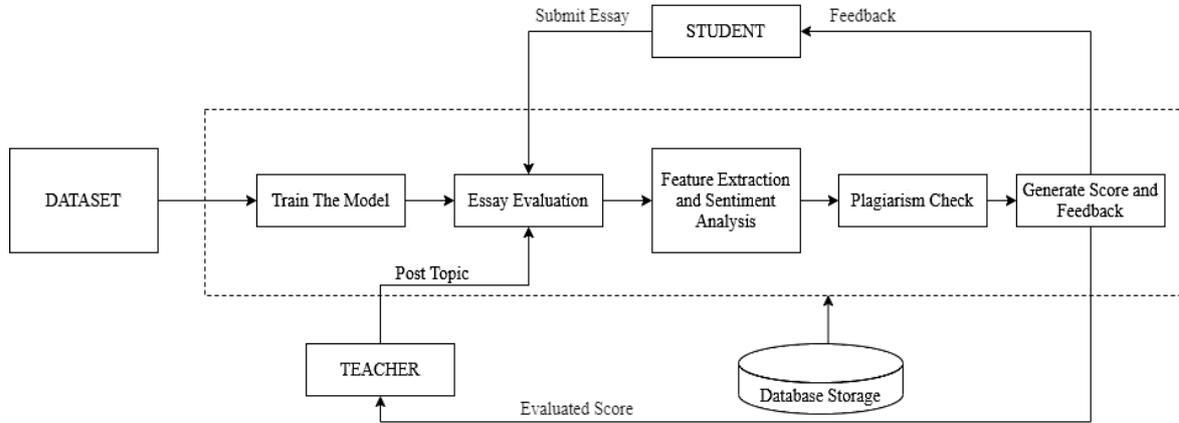


Figure 1. Architecture diagram of automated essay grader

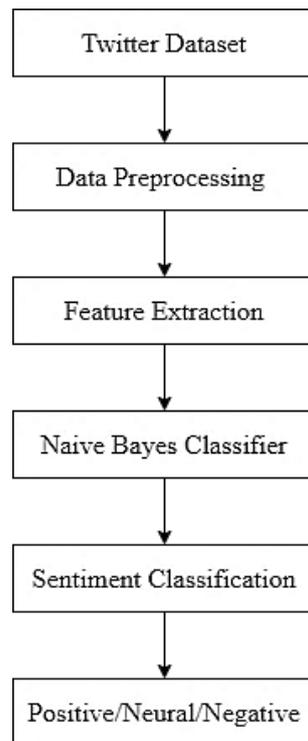


Figure 2. Flow of sentiment analysis

2.6. Essay grading

In this phase the essay submitted by the student is pre-processed by removing stop words and converting the essay from word to vector format. Spelling mistakes and the grammatical errors are detected using the python language tool and the detailed description of each grammatical and spelling error is notified to the student. It first loads the saved Word2Vec model followed by the creation of a word list, an average input feature vector and finally reshapes the submitted essay. It detects the syntactic errors in that essay and appropriately removes the noise from it. It then obtains the result of sentiment analysis followed by the total plagiarism percentage and awards the final marks accordingly [19], [20].

2.7. Generating the corrected essay

In this stage the input essay is checked and all the spelling mistakes, linguistic mistakes and grammatical errors are remedied giving the rethought and amended essay to the student. It first checks the syntactical mistakes in the essay and if it finds any it replaces that mistake with its appropriate correction. Then it creates a new string of text based on the values in the list and the corrected list by joining them [20].

2.8. User interfaces

The teachers and the students, both have their separate online interfaces. Initially they have to create their respective accounts by signing up with their details such as name, mobile number, email address and password. They can use the registered email address and password as login credentials in the future. Each teacher has a dashboard with the following three tabs: view profile tab to view their personal information, create test tab to create a test by mentioning the topic of the essay, the duration of the test and the expected sentiment as either positive, negative or neutral and the view results tab to view the score of each individual student identified with their respective email addresses. Similarly, each student has a dashboard with the following three tabs: view profile tab to view their personal information, take test tab to attempt a test by writing an essay corresponding to the provided topic the topic and submit it accordingly and a view results tab to view their scores and feedback.

Algorithm 1 shows the proposed algorithm. Algorithm loads the saved Word2Vec model [20], then it creates the word list, average input feature and reshapes the submitted essay. It then detects the syntactic errors in that essay and removes the noise from it. It then obtains the sentimental analysis and calculates the total plagiarism and deducts marks accordingly. The overall grade is calculated based on the quality of the essay through feature extraction, the number of syntactic errors, the percentage of plagiarism found and sentiment of the essay [21].

Algorithm 1. Essay grading

Input: Essay written by a student

Output: A score in scale 1-10

```

1: Initialize number of features to 300;
2: Load the weights from the saved model;
3: Load the saved Word2Vec model;
4: Create a word list, average input features and reshape the input essay;
5: Parse the input essay to detect syntactic errors;
6: Create custom tokens for the input essay by removing noise;
7: Obtain sentiment of the essay from the classifier;
8: Calculate total plagiarism;
9: Obtain grade for quality of the essay and reduce it to ratio of 3 from ratio of 10;
10: Determine number of grammatical and spelling mistakes, and deduct 0.25 marks for each;
11: If total plagiarism<15% then
12:   marks=marks;
13: Else If total plagiarism is between 15 to 25% then
14:   marks=marks-0.5;
15: Else If total plagiarism is between 25 to 50% then
16:   marks=marks-1;
17: Else If total plagiarism is between 50 to 75% then
18:   marks=marks-1.5;
19: Else If total plagiarism is between 75 to 85% then
20:   marks=marks-2;
21: Else If total plagiarism is between 85 to 95% then
22:   marks=marks-2.5;
23: Else If total plagiarism>95% then
24:   marks=marks-3;
25: If the obtained sentiment matches the expected sentiment for the topic, then
26:   marks=marks;
27: Else If the expected sentiment is neutral then
28:   marks=marks;
29: Else
30:   marks=marks-1;
31: Round up the marks for quality of the essay up to 2 decimal places;
32: Display comments for the corresponding sentiments of the essay;
33: Calculate the total score;
34: Determine the final score by rounding up the score to the next whole number;
35: If total score is<5 then
36:   performance is poor;
37: Else If score is between 5 and 8 then
38:   Performance is average;
39: Else
40:   Performance is Excellent;

```

3. RESULTS AND DISCUSSION

3.1. Kappa evaluation

The main criteria for evaluating automated essay grading systems (AEGS) is the kappa score, there are two kinds of kappa, linear weighted kappa (LWK) and quadratic weighted kappa (QWK) [21], [22]. These are the error matrices which estimate the extent of similarity between two grades.

$$k = \frac{X_o - X_e}{1 - X_e} \quad (1)$$

$$k = 1 - \frac{1 - X_o}{1 - X_e} \quad (2)$$

In (1) and (2) X_o is the original consistency rate and X_e is the theoretical consistency rate. X_o can be determined by dividing the number of correct groupings for each class with the total number. The kappa score generally lies between 0 and 1. Table 1 shows the physical representations of various kappa values. If the metric is below 0, it indicates that there is less agreement between the raters. The QWK is computed between the automatically generated score and the resolved score of human raters on each set of essays [21], [22]. The QWK is expressed as (3).

$$W_{i,j} = \frac{(i-j)^2}{(N-1)^2} \quad (3)$$

Table 1. The corresponding physical representation of kappa

Kappa value	Physical presentation
0.0-0.20	Almost inconsistent
0.21-0.40	Generally consistent
0.41-0.60	Probably consistent
0.61-0.80	Highly consistent
0.81-1.0	Completely consistent

3.2. Naïve Bayes classification evaluation

Multinomial naïve Bayes grouping calculation remains a standard solution to perform sentiment analysis. The essential thought of naïve Bayes strategy is to discover the probabilities of categories related to messages by utilizing the joint probabilities of classes and words. It is given by (4).

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \quad (4)$$

In (4), $P(y|x)$ is the posterior probability of class (y) given predictor (x), $P(y)$ is the prior probability of the class, $P(x)$ is the prior probability of the predictor and $P(x|y)$ is the likelihood which is the probability of predictor for the given class. The dependent feature vector (x_1, x_2, \dots, x_n) and the class Y being given. Bayes' theorem is expressed in mathematical form as (5).

$$P(Y_j | x_1, \dots, x_n) = \frac{P(x_1, \dots, x_n | Y_j)P(Y_j)}{P(x_1, \dots, x_n)} \quad (5)$$

As per the "Naïve" conditional independence assumption, for the provided class C_k each element of vector x_i is restrictively autonomous of each and every other component x_j for $i \neq j$.

$$P(x_i | Y_j, x_1, \dots, x_n) = P(x_i | Y_j) \quad (6)$$

The simplified relation is stated as (7).

$$P(Y_j | x_1, \dots, x_n) = \frac{P(Y_j) \prod_{i=1}^n P(x_i | Y_j)}{P(x_1, \dots, x_n)} \quad (7)$$

Since $P(Y_j | x_1, \dots, x_n)$ is constant and if the feature variable values are known the classification is written as (8), (9),

$$P(Y_j | x_1, \dots, x_n) = P(Y_j) \prod_{i=1}^n P(x_i | Y_j) \quad (8)$$

$$\hat{x} = \operatorname{argmax}_{j=i..m} P(Y_j) \prod_{i=1}^n P(x_i | Y_j) \quad (9)$$

The various forms of naive Bayes classifiers differ from each other based on the assumptions they make related to the distribution of $P(x_i | Y_j)$. Where $P(Y_j)$ is generally defined to be the relative frequency of class Y_j in the training dataset.

3.3. Experiment setup

LSTM organizations-are considered to be an extraordinary form of recurrent neural network (RNN), used for learning large haul dependencies. Remembering data for considerable stretches of time is essentially their importance. All RNNs possess a type of chain of rehashing modules of neural network. In regular RNNs, this rehashing module has an exceptionally basic design, for example, a solitary tanh layer [23].

$$z_t = LSTM(z_{t-1}, y_t) \quad (10)$$

Where y_t and z_t depict the input vectors at the time t . The LSTM model is expressed in terms of input, output and forget gates, that controls the flow of information within the recursive operation [23]. The function of LSTM model is expressed in the (11)-(16) [24],

$$a_t = \sigma(W_a \cdot y_t + U_a \cdot z_{t-1} + c_a) \quad (11)$$

$$q_t = \sigma(W_q \cdot y_t + U_q \cdot z_{t-1} + c_q) \quad (12)$$

$$r_t = \tanh(W_r \cdot y_t + U_r \cdot z_{t-1} + c_r) \quad (13)$$

$$r_t = a_t \cdot r_t + q_t \cdot r_{t-1} \quad (14)$$

$$b_t = \sigma(W_b \cdot y_t + U_b \cdot z_{t-1} + c_b) \quad (15)$$

$$z_t = b_t \cdot \tanh(r_t) \quad (16)$$

In (10)-(16) input gate a_t , output gate q_t , memory cell r_t , forget gate b_t , and hidden state z_t is needed. Furthermore, each of variable is weighted the input vector y_t , W , U and b are weight matrices and bias vector parameter. Moreover, also used sigmoid function σ , tanh function \tanh . At each time step t , LSTM produces a hidden vector output z_t that affects the semantic representation of the essay residing at position t . The ultimate representation of the essay is again subjected to feature extraction in the self-information layer. We have trained two models for predicting grades and to perform sentiment analysis for an essay submission. The grade prediction model was trained on 12,000 essays belonging to 8 different contexts with each essay having manual grades graded by two teachers. The sentiment analysis model was trained on twitter dataset containing positive and negative samples of data.

3.4. Result

LSTM model was developed utilizing TensorFlow and wrapped with Python library [25]. The automatic essay grading system is fabricated utilizing LSTM model and after training the model using K-fold strategy for 5 folds with 10 epochs each it gained a quadratic weighted kappa score of 0.911. Figure 3 shows the quadratic weight kappa score in each fold.

We can estimate that overall system yields extremely good results for sentiment analysis, grading the essay, plagiarism detection and generating corrected essay. Grades for the essays is provided out of 10 by considering sentiment analysis, plagiarism check and syntactic errors. Essay provided as input to the essay grading model. Once the essay is submitted, it is subjected to the detection of syntactic mistakes such as spelling and grammatical errors. The essay is then checked for plagiarism. If plagiarism is found, the plagiarized URL along with the corresponding plagiarism percentage is retrieved. It also gives the total plagiarism percentage that is necessary to award the final grade.

Now, the essay is subjected to sentiment analysis with the help of the classifier. The expected sentiment and the detected sentiment are compared and marks are deducted if they are not identical. Based on these parameters a final grade along with a comment is provided to the student. As a part of feedback, the students are also provided with the corrected essay. It is the essay written by the student, but without any errors.

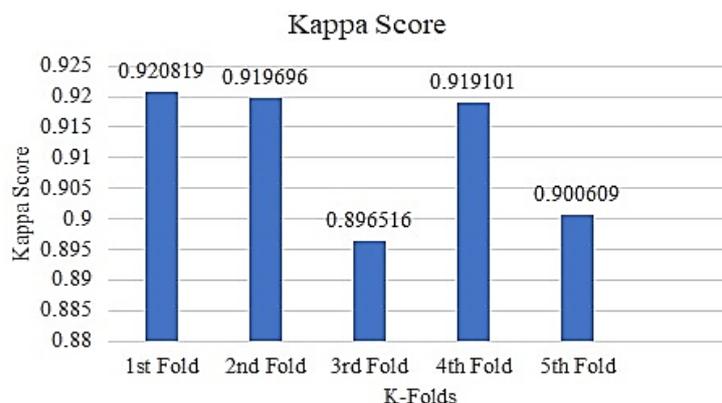


Figure 3. Kappa score collected from each fold

4. CONCLUSION

In this paper, we have presented a model that uses NLP and machine learning to help educational institutions award grades to students based on their essay submissions to produce much accurate and unbiased results with less time consumption. Based on their advantages such models are thought to have a great potential in the future in various fields. This system leverages an LSTM network model. The model was trained with nearly 12,000 essays where each essay was graded by two manual graders. It was trained using K-fold mechanism for 5 folds and each fold operating for 10 epochs each with a batch size of 64. The model was then evaluated using QWK metric that depicted an average score of 0.911. A naïve bayes classifier was employed to build a sentiment classifier to determine either the positive or negative approach of students towards a topic. This classifier showed an accuracy of 99.4%. Along with which the system also detects syntactic errors in the essay such as spelling or grammatical mistakes and provides detailed feedback regarding the same to the students for further corrections.

A plagiarism detector was built to determine if the student has plagiarized any content from the open web sources. If so, the plagiarized URL along with the percentage of plagiarism is depicted. All the above parameters that include number of syntactic mistakes, amount of plagiarism, and the quality of the essay are considered to award the final grade to the student. As a part of maintenance and to improve the performance of the model further gradually, each essay submitted to the essay scoring system is appended to the dataset and the model will be trained on a periodic basis. This would further enhance the accuracy and reliability of the system. Future scope of this system involves exploring much enhanced characteristics to improve the results and make the system more advance. A student's fluency in English can be attempted to judge through the vocabulary and style of language adopted while writing the essay. The feedback features can be further exploited to provide a much valuable and meaningful comments to students for their better performance in the future.

ACKNOWLEDGEMENTS

We would like thank Karnataka State Council of Science and Technology (KSCST) for funding this research.

REFERENCES

- [1] A. Sharma and D. B. Jayagopi, "Automated grading of handwritten essays," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Aug. 2018, pp. 279–284, doi: 10.1109/ICFHR-2018.2018.00056.
- [2] S. Ghosh and S. S. Fatima, "Design of an automated essay grading (AEG) system in Indian context," in *TENCON 2008-2008 IEEE Region 10 Conference*, Nov. 2008, pp. 1–6, doi: 10.1109/TENCON.2008.4766677.
- [3] Z. Wang, J. Liu, and R. Dong, "Intelligent auto-grading system," in *2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, Nov. 2018, pp. 430–435, doi: 10.1109/CCIS.2018.8691244.
- [4] R. G. Patil and S. Z. Ali, "Approaches for automation in assisting evaluator for grading of answer scripts: a survey," in *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, Dec. 2018, pp. 1–6, doi: 10.1109/CCAA.2018.8777664.
- [5] G. F. Fragulis, M. PapatSIMOULI, L. Lazaridis, and I. A. Skordas, "An online dynamic examination system (ODES) based on open source software tools," *Software Impacts*, vol. 7, Feb. 2021, doi: 10.1016/j.simpa.2020.100046.
- [6] G. R. Perera, D. N. Perera, and A. R. Weerasinghe, "A dynamic semantic space modelling approach for short essay grading," in *2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTER)*, Aug. 2015, pp. 43–49, doi: 10.1109/ICTER.2015.7377665.

- [7] B. S. J. Kapoor, S. M. Nagpure, S. S. Kolhatkar, P. G. Chanore, M. M. Vishwakarma, and R. B. Kokate, "An analysis of automated answer evaluation systems based on machine learning," in *2020 International Conference on Inventive Computation Technologies (ICICT)*, Feb. 2020, pp. 439–443, doi: 10.1109/ICICT48043.2020.9112429.
- [8] A. Satyanarayana, R. Lansiquot, and C. Rosalia, "Using prescriptive data analytics to reduce grading bias and foster student success," in *2019 IEEE Frontiers in Education Conference (FIE)*, Oct. 2019, pp. 1–5, doi: 10.1109/FIE43999.2019.9028663.
- [9] Y. M. Arikat, "Subtractive neuro-fuzzy modeling techniques applied to short essay auto-grading problem," in *2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, Mar. 2012, pp. 889–895, doi: 10.1109/SETIT.2012.6482032.
- [10] J. Hoblos, "Experimenting with latent semantic analysis and latent dirichlet allocation on automated essay grading," in *2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS)*, Dec. 2020, pp. 1–7, doi: 10.1109/SNAMS52053.2020.9336533.
- [11] D. Zhang and X. Yuan, "Intelligent scoring of English composition by machine learning from the perspective of natural language processing," *Mathematical Problems in Engineering*, vol. 2022, pp. 1–9, May 2022, doi: 10.1155/2022/9070272.
- [12] A. R. Razon, M. L. J. Vargas, R. C. L. Guevara, and P. C. Naval, "Automated essay content analysis based on concept indexing with fuzzy c-means clustering," in *2010 IEEE Asia Pacific Conference on Circuits and Systems*, Dec. 2010, pp. 1167–1170, doi: 10.1109/APCCAS.2010.5775058.
- [13] A. A. P. Ratna, H. Artajaya, and B. A. Adhi, "GLSA based online essay grading system," in *Proceedings of 2013 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, Aug. 2013, pp. 358–361, doi: 10.1109/TALE.2013.6654461.
- [14] S. M. F. D. S. Mustapha, N. Idris, and R. Abdullah, "Embedding information retrieval and nearest-neighbour algorithm into automated essay grading system," in *Third International Conference on Information Technology and Applications (ICITA'05)*, 2005, vol. 2, pp. 169–172, doi: 10.1109/ICITA.2005.126.
- [15] H. K. Janda, A. Pawar, S. Du, and V. Mago, "Syntactic, semantic and sentiment analysis: the joint effect on automated essay evaluation," *IEEE Access*, vol. 7, pp. 108486–108503, 2019, doi: 10.1109/ACCESS.2019.2933354.
- [16] M. M. Islam and A. S. M. L. Hoque, "Automated essay scoring using generalized latent semantic analysis," in *2010 13th International Conference on Computer and Information Technology (ICCIT)*, Dec. 2010, pp. 358–363, doi: 10.1109/ICCITECHN.2010.5723884.
- [17] A. Olowolayemo, S. D. Nawi, and T. Mantoro, "Short answer scoring in english grammar using text similarity measurement," in *2018 International Conference on Computing, Engineering, and Design (ICCED)*, Sep. 2018, pp. 131–136, doi: 10.1109/ICCED.2018.00034.
- [18] J. Lemantara, M. J. Dewiyani Sunarto, B. Hariadi, T. Sagirani, and T. Amelia, "Prototype of automatic essay assessment and plagiarism detection on mobile learning 'Molearn' application using GLSA method," in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Dec. 2019, pp. 314–319, doi: 10.1109/ISRITI48646.2019.9034652.
- [19] S. V. Shetty, K. R. Guruvyas, P. P. Patil, and J. J. Acharya, "Essay scoring systems using AI and feature extraction: a review," in *Proceedings of Third International Conference on Communication, Computing and Electronics Systems*, Springer Singapore, 2022, pp. 45–57, doi: 10.1007/978-981-16-8862-1_4.
- [20] D. Ramesh and S. K. Sanampudi, "An automated essay scoring systems: a systematic literature review," *Artificial Intelligence Review*, vol. 55, no. 3, pp. 2495–2527, Mar. 2022, doi: 10.1007/s10462-021-10068-2.
- [21] Y. Salim, V. Stevanus, E. Barlian, A. C. Sari, and D. Suhartono, "Automated English digital essay grader using machine learning," in *2019 IEEE International Conference on Engineering, Technology and Education (TALE)*, Dec. 2019, pp. 1–6, doi: 10.1109/TALE48000.2019.9226022.
- [22] M. Injadat, A. Moubayed, A. B. Nassif, and A. Shami, "Machine learning towards intelligent systems: applications, challenges, and opportunities," *Artificial Intelligence Review*, vol. 54, no. 5, pp. 3299–3348, Jun. 2021, doi: 10.1007/s10462-020-09948-w.
- [23] A. Wiratmo and C. Fatichah, "Assessment of Indonesian short essay using transfer learning siamese dependency tree-LSTM," in *2020 4th International Conference on Informatics and Computational Sciences (ICICoS)*, Nov. 2020, pp. 1–5, doi: 10.1109/ICICoS51170.2020.9299044.
- [24] Y. Wang, Z. Wei, Y. Zhou, and X. Huang, "Automatic essay scoring incorporating rating schema via reinforcement learning," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 791–797, doi: 10.18653/v1/D18-1090.
- [25] A. Shehab, M. Elhoseny, and A. E. Hassanien, "A hybrid scheme for automated essay grading based on LVQ and NLP techniques," in *2016 12th International Computer Engineering Conference (ICENCO)*, Dec. 2016, pp. 65–70, doi: 10.1109/ICENCO.2016.7856447.

BIOGRAPHIES OF AUTHORS



Vijaya Shetty Sadanand      is currently working as Professor in the Department of Computer Science and Engineering, at Nitte Meenakshi Institute of Technology, Bangalore. Her research interests include data mining, machine learning, deep learning and distributed computing. She is a Life member of Indian Society for Technical Education (ISTE) and member of Computer Society of India (CSI). She can be contacted at email: vijayashetty.s@nmit.ac.in.



Kadagathur Raghavendra Rao Guruvyas    is an undergraduate student pursuing Computer Science and Engineering at Nitte Meenakshi Institute Of Technology, Bangalore. He has worked on various data science and machine learning related projects and successfully completed two internships at Raman Research Institute. Moreover, he is a data scientist enthusiast and excited to solve various problems. He can be contacted at email: krgvyas@gmail.com.



Pranav Prashantha Patil    is an undergraduate student pursuing Computer Science and Engineering at Nitte Meenakshi Institute Of Technology, Bangalore. Pranav has worked on various Full stack based projects and successfully completed one year internship at Actyv. He is a data science enthusiast and currently gaining experience at Actyv. Apart from this he is a tech geek and is excited for new challenges. He can be contacted at email: pranavppatil08@gmail.com.



Jeevan Janardhan Acharya    is an undergraduate student pursuing Computer Science and Engineering at Nitte Meenakshi Institute of Technology, Bangalore. He has worked on various cloud based project such as AWS Environment Data Analysis, Covid-19 assistance chatbot using AWS. Bangalore. He is currently a fresher gaining experience at Larsen and Tourbo Technology Services (LTTS). Besides, he has actively participated in Unnat Bharat Abhiyan as a student volunteer. He can be contacted at email: jeevanjacharya04@gmail.com.



Sharvani Gunakimath Suryakanth    is an Associate Professor with the Department of Computer Science and Engineering at R. V. College of Engineering, Bengaluru, Karnataka, India. Her areas of specialisation are cloud computing, machine learning, internet of things and sensor networks. She can be contacted at email: sharvanigs@rvce.edu.in.