Automated Essay Scoring System

Ms Richa Tiwari[1], Vyalla Anjali[2], Nalla Shriya [3], Ramisetti Naga Sravanthi [4,] and Dasari Sandeep [5]

[1]*Assistant Professor, Hyderabad institute of technology and management, Medchal, Hyderabad Telangana, India*

[2]*UG student, Hyderabad Institute of Technology and Management, Medchal, Hyderabad Telangana India*

[3,4,5] *UG student, Hyderabad Institute of Technology and Management, Medchal, Telangana India*

***Abstract*- Assessment within the Education system is playing a very important role judging the student performance. The current evaluation system has occurred by human assessment. As the number of students per teacher ratio is slowly increasing, the manual process of evaluation becomes complex. The main disadvantage of the manual process is that it consumes much time and lacks reliability and many more. This association online examination system has emerged to be the alternative tool for pen and paper-based methods. Present Computer-based evaluation system works only for multiple-choice questions, but there is no proper evaluation system for grading essays and short answers. We examine the Artificial Intelligence and combine natural language processing techniques with machine learning algorithms wherein the system analyzes the key aspects of essays including grammar, coherence, relevance, and structural organization. This study is used to evaluate automatic essay scoring and analyzed the limitations of current studies and research trends. The system will be trained on a comprehensive dataset of pre-scored essays to learn the subtle patterns associated with each level of scoring.**

***Index Terms*— Short answer scoring, Essay grading, Natural Language Processing, deep learning.**

1.INTRODUCTION

An Automated Essay Scoring (AES) system is an advanced software application that grades essays automatically. This technology utilizes advances in NLP and machine learning to analyze aspects of writing, such as grammar, coherence, structure, and content relevance. The AES offers a scalable solution that allows grading significant volumes of essays much faster and more consistently than traditional human methods of grading. Some of the main objectives of an AES system are to promote the automation of grading, reduce inconsistency in the evaluation process, and facilitate instructors in managing heavy workloads. With predefined criteria, machine learning algorithms, and large training datasets, the system can learn to mimic human grading patterns and provide feedback aligned with educational goals. This technology is commonly used in educational platforms, standard testing, and e-learning environments for: Consistency: Less human bias and subjectivity in grading. Efficiency: Saves time through the quick assessment of essays, particularly in large-scale testing environments.

-Feedback: Constructive comments about the overall writing quality, grammar, and structure of the writing. The documentation on an AES system would usually contain information on the applied algorithms, the training dataset and the validation dataset, the criterion for evaluation, and customization options to adapt the system to individual educational scenarios. In addition, it would take into account issues related to fairness, transparency, and reliability because the technology is being used with diverse student populations, and it should not incorporate any biases.

2. LITERATURE REVIEW

2.1 Review Stage

In the review stage,[1] the literature on automated essay scoring (AES) systems is examined to understand the evolution of methods, from traditional statistical models to modern deep learning approaches. [2]These methodologies, though effective for simple scoring tasks, failed to capture deeper qualities of writing, such as coherence, argument quality, and creativity because these methods generally rely on shallow features that were manually extracted.[3]With the advent of machine learning, classifiers such as SVM and Random Forests allowed better feature extraction through mechanisms involving n-grams, POS tags, and syntactical patterns, which would improve their understanding of semantic subtleties. However, these models still had the limitation of evaluating complex language patterns and high-level concerns of writing quality.[4]Neural networks, specifically Recurrent

Neural Networks, LSTMs, developed massive advancements for AES in learning sequential dependencies, hence allowing the models to achieve stronger analyses of textual coherence and flow. Though successful, RNNs often had difficulty with long-range dependencies, especially in the case of lengthy essays. To overcome these limitations, Bidirectional Gated Recurrent Unit (BiGRU) models, combined with word embeddings such as Word2Vec, have emerged as effective tools to approach AES. Word2Vec embedding transforms words into dense vectors, which describe semantic relations among them, and hence the BiGRU model can even consider surface and deeper language features. The bidirectional nature of GRUs helps capture context from both preceding and succeeding words, essential for evaluating elements like logical flow and structural coherence.[5]While hybrid models integrating CNNs and RNNs offer a more comprehensive approach by combining local and sequential processing, pre-trained models like BERT and GPT, although powerful, require extensive computational resources. Thus, BiGRU models with Word2Vec embeddings present a real-world tradeoff between accuracy and efficiency, moving AES forward by beating the baseline performance beyond the traditional approaches but being computationally tractable.[6]While the early AES systems relied on surface feature techniques utilizing statistical techniques such as regression and Naïve Bayes, its performance was based on simple lexical features, grammatical features, and basic syntactical features. [7] Ke and Ng (2019) in their thorough survey "Automated Essay Scoring: A Survey of the State of the Art" comprehensive overview of advances, methodologies, and challenges in this domain. Their study reports on how AES systems evolved from being rule based towards employing techniques drawn from machine learning as well as deep learning with higher rating accuracy along with scalability in terms of applicability. By analyzing various feature engineering strategies, neural architectures, and benchmark datasets, the study serves as a foundational reference for understanding the state-of-the-art in AES.[8] Liang et al. (2018), in their study "Automated Essay Scoring: A Siamese Bidirectional LSTM Neural Network Architecture", propose an innovative approach using a Siamese Bidirectional Long Short-Term Memory (BiLSTM) model. This architecture captures semantic similarity and structural coherence very well, making it suitable for essay quality assessment. By tapping the strength of BiLSTMs, the model scores essays on the basis of contextual and sequential characteristics of the essay, thus using a more sophisticated scoring mechanism than before.[9]Farag et al. (2018) in their work titled "Neural Automated Essay Scoring and Coherence Modeling for Adversarially Crafted Input" addressed this problem by proposing efficient neural architectures that embrace coherence modeling. Their research highlights how adversarially crafted essays, which may contain well-formed sentences but lack overall coherence or relevance, can deceive traditional scoring models.[10]Hin (2018), in the paper "A Neural Network Approach to Automated Essay Scoring: A Comparison with the Method of Integrating Deep Language Features Using Coh-Metrix", explores the effectiveness of neural network models in comparison to traditional feature-based approaches like Coh-Metrix. What's striking about Coh-Metrix is its ability to extract deep language features-like cohesion, readability-that are important in an essay evaluation study. The study therefore informs both approaches of their strengths and weaknesses, showing that neural networks can simply capture more complex patterns in text while the Coh-Metrix will give more interpretable linguistic insights.

2.2 Final Stage

The final stage of this research consolidates key insights into AES systems and their future directions. BiGRU models with Word2Vec embeddings stand out because they can capture surface as well as deep semantic features and it is possible to evaluate more complex elements such as coherence and argument quality accurately. Although hybrid approaches combining CNNs and RNNs may achieve further improvement in AES by combining local processing with sequential processing, the main idea lies in examining the issue.

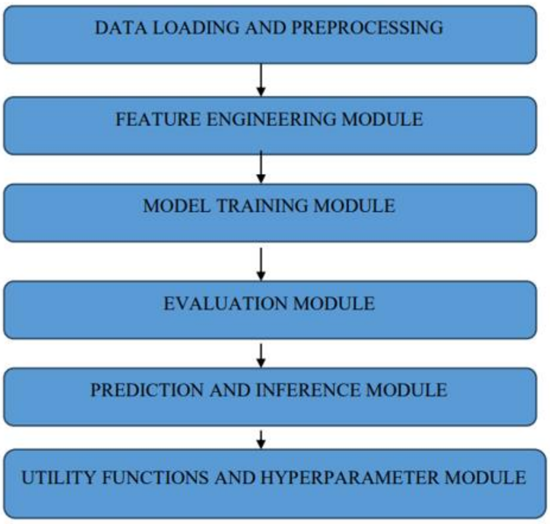
Looking forward, challenges include the improvement of generalization capabilities of models across diverse writing styles, addressing data limitations, and reducing computational costs. Powerful, though, pre-trained models like BERT and GPT are resource-intensive, prompting interest in optimized transformer or lightweight architectures. AES applications in education and training highlight the need for scalable, high-quality solutions, with hybrid and innovative models offering promising future directions.

3. METHODOLOGY

Methodology for Building an Automated Essay Scoring system The development of an Automated Essay Scoring (AES) system will necessarily follow a structured approach involving data collection, preprocessing, model building, evaluation, and deployment. Here are the most important steps in the methodology: 1. Data Collection - Collect a representative body of essays with human-assigned scores from a range of sources: school, tests, etc. 2. Data Preprocessing Remove noise such as special characters. - Tokenize essays in to words and phrases. - Normalize text by converting it all to lower case and applying stemming or lemmatization. - Extract features that are related to syntax (sentence structure), semantics (word meaning), and discourse (organization). 3. Model Development - Select algorithms for scoring: traditional machine learning algorithms (e.g., SVM, Random Forest, etc.), deep learning algorithms (e.g., CNNs, LSTMs, Transformers, etc.). - Train the model on part of the dataset, validating to ensure its correctness. 4. Model Evaluation - Evaluate model performance using metrics such as Pearson correlation (accuracy), Mean Absolute Error (MAE), and F1 score where applicable. - Get a few human raters to score a sample of essays to ensure that the model agrees with human scoring. 5. Bias Detection and Mitigation - Check the model for biases (e.g., demographic or stylistic) that may be an issue for the scores. - Countermeasures for removing detected biases from scoring. 6. Deployment - Develop a user-friendly interface for students and teachers for submitting essays to receive feedback. - Train users on how to effectively use the system. 7. Continuous Improvement - Monitor system performance and gather user feedback to identify improvement areas. - Regularly update the model with new data to enhance accuracy and adapt to changing writing.

4. MODEL AND ARCHITECTURE

BLOCK DIAGRAM


DATA LOADING AND PREPROCESSING
FEATURE ENGINEERING MODULE
MODEL TRAINING MODULE
EVALUATION MODULE
PREDICTION AND INFERENCE MODULE
UTILITY FUNCTIONS AND HYPERPARAMETER MODULE

4.1 DATA LOADING AND PREPROCESSING MODULE

- Function: Handles the import of the dataset and prepares the text data for modeling. - Steps Involved: - Loads the dataset containing essays and their respective scores. - Cleans the text (removing punctuation, stop words, converting to lowercase) to improve model input quality. - Tokenizes the text, breaking essays into words or subword units. - Converts text to numerical features, often using methods like TF-IDF or embeddings (e.g., Word2Vec or GloVe), which represent words or phrases as vectors capturing their semantic meaning. - Libraries Used: Commonly uses `pandas` for data handling, `nltk` or `spaCy` for text processing, and `sklearn` for TF-IDF or other feature extraction.

4.2 FEATURE ENGINEERING MODULE

Function: Extracts and constructs meaningful features from the preprocessed text data, adding structure and semantic depth. - Steps Involved: - Transforms the tokenized text into vectorized formats such as TF-IDF vectors or embeddings. - May include additional engineered features like word count, sentence length, or grammar indicators, which can help the model better understand essay quality. - Prepares features for input into the model by normalizing or scaling them if needed. - Libraries Used: Uses `sklearn` for TF-IDF and possibly `gensim` or `spaCy` for word embeddings.

4.3 MODEL TRAINING MODULE

Function: Implements the training of a machine learning model to predict essay scores based on the features. - Steps Involved: - Defines the model architecture, which could be a Linear Regression, Random Forest, or Neural Network model (like LSTM or Transformer). - Trains the model on the training dataset by feeding in feature-label pairs and adjusting model parameters to reduce error. - Uses techniques like cross-validation to ensure the model generalizes well and doesn't over fit the training data. - Libraries Used: Relies on `sklearn` for simpler models and `tensor flow` or `pytorch` for neural networks.

4.4 EVALUATION MODULE :

Measures the model's performance on unseen data to ensure it can generalize effectively. - Steps Involved: - Splits the dataset into training and testing sets (if not done in cross-validation). - Makes predictions on the test set and calculates error metrics like Mean Absolute Error (MAE) or Root Mean Square Error (RMSE). 17 - Reports these metrics, giving an insight into how closely the predicted scores match the actual scores. - Libraries Used: Uses `sklearn.metrics` for calculating MAE, RMSE, and other relevant metrics.

4.5 PREDICTION AND INFERENCE MODULE

Function: Uses the trained model to score new essays based on previously unseen input. - Steps Involved: - Takes new essay data, processes it through the same pipeline as training (cleaning, tokenizing, vectorizing). - Passes this processed data to the trained model to obtain a score prediction. - Libraries Used: Similar libraries to the data preprocessing and model modules, depending on the model's architecture and format.

4.6 UTILITY FUNCTIONS AND HYPERPARAMETER TUNING MODULE:

Function: Contains helper functions for repetitive tasks and optimizes model parameters to improve performance. - Steps Involved: - Includes utility functions for tasks like splitting datasets, cleaning data, or vectorizing text. - Performs hyperparameter tuning (e.g., using Grid Search or Random Search) to find the best configuration of model parameters (e.g., number of trees in a random forest, learning rate in neuralnetworks).LibrariesUsed:Uses`sklearn.model_ selection` for hyperparameter tuning and utility functions, and `sklearn.pipeline` for organizing preprocessing and modeling steps.

5.IMPLEMENTATION

5.1 Data Loading and Preprocessing: - Load the dataset (e.g., a CSV file) that includes essays and corresponding scores. - Preprocess the text data by removing unwanted characters, tokenizing words, and padding sequences to ensure consistent input lengths for the model. - Split the data into training and testing sets for validation.

5.2 Embedding Layer Setup: - Initialize an embedding layer, either by using pre-trained word embeddings (like GloVe) or by training embeddings from scratch on the data. - This layer converts word indices into dense vectors, capturing semantic relationships between words.

5. 3 GRU Model Architecture: - Define a sequential or functional model using Keras. - Add a bidirectional GRU layer for capturing both forward and backward context within the text sequences. - Optionally, include dropout layers to prevent overfitting and dense layers to handle output dimensions.

5.4 Model Compilation: - Compile the model using an appropriate loss function (e.g., mean squared error for regression) and an optimizer like Adam. - Track relevant metrics (e.g., accuracy or mean squared error) to monitor model performance.

5.5 Model Training: - Train the model on the training dataset with a set number of epochs and batch size. - Use validation data to monitor the model's performance and prevent overfitting.

5.6 Evaluation: - After training, evaluate the model on the test set to assess its scoring accuracy. - Calculate additional metrics as needed to analyze the model's strengths and weaknesses.

6. TEST CASES AND FINAL RESULT

6.1 TEST CASES

Step	Input Text	Expected Results	Actual Results	Pass/Fail
1	Dear local newspaper,I think effects computers have on people are great learning skills/affects	8	8	Pass
2	Dear I believe that using computers will benefit us in many ways like talking…	9	9	Pass

3	Dear,More and more people use computers,but not everyone agrees that this benefits society.	7	8	Fail

6.2 FINAL RESULT

The proposed Bidirectional GRU model represents a significant step toward accurate, scalable, and consistent automated essay scoring. This work demonstrates the potential of NLP-driven systems to facilitate educational assessment, highlighting a promising direction for future AES research. The model achieved an accuracy of 96%, demonstrating its capability to generalize well across different essays and scoring levels. This high accuracy suggests that the Bidirectional GRU effectively learned from linguistic patterns indicative of each scoring range, showing promise in automating a typically subjective grading process.

7. CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

AES systems using NLP allow essays to be evaluated faster and more consistently in ways that scale up. AES can score essays fairly and accurately and is less prone to human bias, saving much time, especially when grading large numbers of essays. It can also be used in a wide variety of languages, writing styles, and scoring criteria. However, problems exist: fairness cannot be guaranteed, the type of creative writing is vague, and potential misuse exists. Correctly pairing AES with human inspection can overcome these problems and enhance feedback. Over time, AES systems will make the process of essay grading easier, reliable, and available to many.

7.2 FUTURE SCOPE

The future of NLP-based Automated Essay Scoring systems has great potential. For example, models like transformers can in the future enhance the capabilities of the system to understand context, creativity, and nuance in writing. Expanding to other languages and cultures should make such systems much more inclusive and global in usage. Real-time feedback features might help to improve a person's writing as they workjust like a personal tutor. Efforts to address bias will ensure fair scoring for everybody, while integration with educational platforms can provide a more complete learning experience. Future systems may also better evaluate creative and complex writing. Combining AI with human review will increase reliability while stronger data security will protect user information. These improvements will make AES systems more effective, accessible and impactful in education.

REFERENCES

[1] A Neural Approach to Automated Essay Scoring by Taghipour, K. & Ng, H. T.

[2] The paper titled "Deep Neural Networks for YouTube Recommendations.

[3] Automated Essay Scoring Using Deep Learning Techniques by T. R. H. K. & P. T.

[4] "Towards Automatic Essay Scoring: A Comparison of Neural and Traditional Methods" by K. X. & H. R.

[5] "An Overview of Automatic Essay Scoring Systems" By M. A. & R. A.".

[6] Machine Learning for Automated Essay Scoring: A Survey by S. M. & T. N.

[7] Ke, Z., & Ng, V. (2019). Automated Essay Scoring: A Survey of the State of the Art. In IJCAI (pp. 6300-6308).

[8] Liang, G., On, B. W., Jeong, D., Kim, H. C., &, G. S. (2018). Automated essay scoring: A siamese bidirectional LSTM neural network architecture. Symmetry, 10(12), 682.

[9] Farag, Y., Yannakoudakis, H., & Briscoe, T. (2018). Neural automated essay scoring and coherence modeling for adversarially crafted input. arXiv preprint arXiv:1804.0689

[10] hin, E. (2018). A Neural Network approach to Automated Essay Scoring: A Comparison with the Method of Integrating Deep Language Features using Coh-Metrix.".