Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 46 No. 12
December 2025

The Implementation of Smart NLP-ML Educational Theoretical Content Grading System

Joseph O. Ezike, Temitope E. Ogunbiyi, Olufemi A. Adekunle, and Millicent N. Ubaka
Department of Computer Science and Information Technology,
Bells University of Technology, Ota, Nigeria

Abstract

Introduction: The education landscape is evolving rapidly with technological advancements, prompting a paradigm shift in assessment practices. Key challenges in the existing literature include the absence of a unified approach to seamlessly integrate features from ML and NLP, and a lack of empirical evaluations comparing the performance of different feature sets.

Objectives: The study focussed on the design of a hybrid grading model that harmoniously combines topic-based and word embedding-based features for the development of the NLP-ML theoretical content grading system coined "SMART-CGS".

Methods: Primary data was obtained from a lecturer of undergraduate studies that contains the responses of students to an open-ended question in the computer science domain. The datasets were preprocessed with Natural Language Toolkit, Bidirectional Encoder Representation from Transformer (BERT) was used for the word embedding while the topic modelling was done with Latent Dirichlet Allocation. Extra-large bidirectional transformer network (XLNET) was used to benchmark BERT. Random Forest (RF) algorithm was used for the grade's prediction. All the implementation was done using Python libraries and codes.

Results: The RF model's predictions generally show a trend of underestimating the actual grades, suggesting a conservative approach in its evaluations. The precision, recall, F1 score, and accuracy all ranging around 0.92, 0.94 for BERT and XLNET respectively, and indicate a relatively high level of overall model performance. While there is a degree of alignment between the actual and predicted grades, it is evident that the XLNET model perform better than BERT. The scalability of BERT and XLNET-based models are 47 and 78 seconds respectively when tested over different students' responses. This shows a high throughput of executing 500 batch sizes in the part of BERT over XLNET.

Conclusions: The observed conservative grading pattern, potentially influenced by the dominance of lower grades in the dataset, suggests the need for ongoing refinement in feature engineering which prompted the usage of XLNET that gives a better result (0.94).With these results, the choice of the model to adopt depends on the speed and accuracy trade-off that existed between the two models.

Keywords: Smart grading system, Natural Language Processing, Latent Dirichlet Allocation, BERT, XLNET, Educational theory content, SMART-CGS.

1. Introduction

The technological advancement powered by Artificial Intelligence (AI) has influenced the educational sector positively which brought a shift in assessment methods. Education effectiveness is measure by assessing and measuring learning retention in students. Summative assessment techniques, such as written tests, are essential instruments for gauging learning results, whereas formative assessment gives students continuous feedback through tasks [1]. Human examiners have been the mainstay of traditional educational assessment, using their expertise and judgement to gauge student achievement. Human assessment is labour-intensive and time-consuming, which is a major disadvantage. In larger educational contexts or when comprehensive feedback is required, human evaluation does not scale effectively. Students' educational processes can become hampered by a delay in feedback since they might not be able to quickly correct their errors or misunderstandings [2]. Many assets are needed for human evaluation, such as

time, effort, and frequently monetary investment in the educating and training of assessors. These needs can be especially difficult to meet in environments with limited resources. Schools and other educational establishments might find it difficult to devote enough funds to uphold a high level of human evaluation, which could result in concessions in the calibre of instruction and student feedback. Furthermore, the ability of instructors to properly gather, evaluate, and prepare in response to both formal and informal evaluation data is typically limited [3]. In addition to taking a lot of time, traditional method of managing massive amounts of data is prone to human error. This restriction has an impact on the tests' accuracy and dependability, which could result in skewed or inaccurate judgements on student performance. Errors are inevitable in manual data analysis and grading procedures. Because of weariness, negligence, or partiality, human assessors can make errors that compromise an evaluation's accuracy and integrity.

All these aforementioned issues can be resolved using an automated content grading system. An automated content grading techniques such as, natural language processing (NLP) and machine learning (ML) have become a hot topic among researchers nowadays [4]. Thankfully, technological developments have led to the introduction of machine evaluations that use artificial intelligence to effectively analyse data and give feedback [2]. Machine learning (ML) aims to teach computers to solve certain issues using data samples or prior knowledge. ML is used in a variety of fields, including image detection, computer vision, bioinformatics, and education [5]. Because education is evolving daily, this study focusses on the application of NLP and ML in content grading. ML is employed, for instance, to assess students' knowledge, gauge their performance, assist teachers, and more. This paper presents a NLP-ML approach to theoretical content grading.

2. Literature Review

The integration of NLP and ML in education is transforming traditional learning paradigms and providing previously unheard-of chances to create individualised, effective, and easily accessed learning environments. AI is emerging as a crucial tool to close these disparities as educational systems around the globe struggle with a variety of student demands, learning styles, and the need for more individualised approaches [6]. Higher levels of engagement and better educational results can be achieved by educators using AI technologies to customise educational activities to each student's unique needs. AI has the capacity to change education in ways that transcend customised instruction. It promises to improve the efficiency of educational processes by streamlining administrative procedures, allocating resources optimally, and providing real-time feedback. Furthermore, AI can greatly improve accessibility, removing obstacles for learners from different linguistic backgrounds, those in remote locations, and students with disabilities. AI's position in education is becoming more and more important as the world transitions to a more technologically advanced and connected future, providing a route to more effective and equitable learning opportunities for everyone. The various applications of AI in education are examined in this chapter [7].

The combination of topic-based modeling and word embedding approach present a strong method in the domain of NLP and ML. There exist differ kinds of topic-based modeling, example include Latent Dirichlet Allocation (LDA), and each is capable of revealing the latent thematic patterns hidden in a corpus thereby enabling ways to the identification of major topics and themes [8]. Its counterpart, word embeddings, examples include Bidirectional Encoder Representation from Transformer (BERT), Extra-large bidirectional transformer network (XLNET), Word2Vec, or GloVe, shows the embedded semantic connection among words in a continuous vector space [9]. The integration of topic-based modeling and word embedding-based method enables the grasping of the thematic and sematic meanings existed within a corpus of textual data. This hybrid model offers an advantage in the presentation of an efficient language representation, and the revelation of higher topical data and complex semantic details. The work of topic modelling is the provision of a categorial comprehension of the content for the identification of key terms or words [8] whereas, word embeddings boost this comprehension through maintenance of the semantic relationships and contextual meanings. The combination of these two approaches establishes a better features of ML models for content grading, sentiment analysis and soon.

This investigation by [10] introduces the Intelligent Natural Language Processing Essay Grading System (iNLPEGS), a scoring system that can handle increasingly challenging questions and has a high

accuracy percentage and minimal loss function. Semantic analysis and Part of Speech labelling were assisted by a secondary dataset that was gathered from Kaggle and made available by The Hewlett Foundation. A collection of computer science questions and responses was gathered from the computer science department at Babcock University in order to build a more reliable dataset. Based on Enhanced Latent Semantic Analysis employing Part of Speech n-gram Inverse Document Frequency, an Intelligent Natural Language Processing Essay Grading Model was created. In recognition of the accessibility of multiple Python libraries plus SQLite as the database, a web-based application was created utilising Django, Gensim, Jupyter Notebook, and Anaconda. The results of the iNLPEGS performance evaluation indicated an accuracy of 89.03% and an error of 10.97%, suggesting that the scores from the designed intelligent essay grading system and a human grader differ very little. Additionally, the Root Mean Square Error (RSME) loss function displayed a value of 0.620, which is extremely low and indicates proximity to the regression equation's line of best fit. The author of [11] observed that a variety of tests are administered yearly, including competitive, intuitive, and non-institutional exams that students apply for. Competitive and entrance exams typically have multiple-choice or objective questions. These tests are simple to evaluate because they are conducted and analysed on the device. However, there is still no way to respond to and assess descriptive questions because these tests only cover multiple-choice questions. Academic institutions will benefit greatly if the process of evaluating descriptive responses is automated to efficiently assess the student's exam response sheets. Therefore, they proposes a novel approach to assess students' brief responses, such as descriptive responses utilising natural language processing [NLP] algorithms. The employee creates a keyword dataset and response sheet for the system's examination procedure. These datasets are kept in data storage, and students fill up the exam site with their responses. This system computes results automatically using NLP methods. The pre-processing technique was used on the student replies prior to this assessment procedure.

Also, the research presented by [12] compares the online English test system based on machine learning with the conventional paper-based English test system and thoroughly examines the ML-based online English test system from a number of angles. Ultimately, it concludes that the machine learning-based online English test system breaks free from the limitations of the conventional paper English test and increases the test's effectiveness. Additionally, it enhances the speed of marking while preserving the fairness of the English test. However, in order to achieve the integrated English examination system, the English online examination system can offer robust online examination assistance for remote teaching, which can be tightly linked with online courses. The authors in [13] suggest a computerised scoring method based on a deep learning model to increase the accuracy of the scoring model for a Chinese essay, and we validate its impact through two comparative experiments. According to the experimental data, the accuracy of the suggested model is much greater than that of multiple linear regression (MLR), which was previously widely utilised. The suggested model's three accuracy rates are similar to those of the inexperienced teacher. In comparison to the novice teacher, the proposed model's correlation coefficient is much greater and its root mean square error (RMSE) is somewhat lower. The work of [14] aimed to determine how well Decision Trees (DTs) mark problems on the Objective Structured Practical Examination (OSPE) in order to develop an intelligent online tutoring system. The study used the final OSPE findings from the anatomy and physiology course in the Faculty of Health Sciences at McMaster University's winter 2020 semester (HTHSCI 2FF3/2LL3/1D06). For each of the 54 questions, a DT was trained using 90% of the data set in a 10-fold validation method. Each DT was made up of distinct terms that might be found in accurate responses provided by students. The generated DTs indicated the remaining 10% of the data set. The DT attained an average accuracy of 94.49% across all 54 questions when the answers marked by the DT were compared to the answers marked by staff and professors. This implies that machine learning algorithms, like DTs, are a very good choice for OSPE grading and can be used to create an intelligent online OSPE teaching system. The authors in [15] implements a new model for automatic long-text analysis in Open and Distance Learning (ODL) using seven machine-learning techniques in order to overcome the aforementioned issues. In order to grade easy-type exams, this study used seven machine learning approaches and conducted a performance-based comparison study. Additionally, a two-top monolingual aligner for text-matching tasks like long-text, textual entailment recognition, and paraphrase

detection was created. According to the results, the Support Vector Machine outperformed the Naïve Bayes classifier in terms of performance accuracy, followed by Artificial Neural Networks, Decision Trees, Logistic Regression, K-Nearest Neighbour, and Random Forests.

Nevertheless, research gaps have been identified through the survey of literature on the educational assessment methods. Theses existing gaps need to be understood so as to proffer appropriate solution for an effective and non-bias grading system. there exist no proper integration of NLP and ML features and absence of empirical analysis for comparing the performance of these sets of features. From the literature, there is a need for a hybrid content assessment system that will capture both the semantic and syntax of the textual data for a better scoring. Therefore, this study bridged this gap by implementation of a hybrid grading system coined Smart Theoretical Content Grading System (SMART-CGS) that harmoniously combines topic-based and word embedding-based features. This system is aimed to assessment essay answer without any prejudice plus lesser time with accurate results.

3. Methods

In this study, seven steps were involved and they are: data acquisition, data preprocessing, word embedding of student responses, topic modelling of student responses, random forest modelling, experimental setting, and performance evaluation. These steps are explained as follows:

Data Acquisition

The primary data employed for this study was obtained from a lecturer of undergraduate studies, which contains the responses of students to an open-ended question in the computer science subject area. The response of each student is labeled with the actual grade, as determined by the lecturer. The responses and the corresponding grading serve as the independent and dependent variables respectively for the supervised machine learning intent of this study. The student responses (answers) serve as inputs into the word embedding and topic modeling of the conceptual framework, which returns numeric vectors and numeric topic weights respectively. The outputs are then concatenated, and subsequently serve as the predictive attributes (independent variables), while the numeric grading is retained as the dependent variable (class). The acquired student responses (answers) are however subjected to various pre-processing stages, to have a refined input into the topic modeling and word embedding phases respectively.

Data Pre-processing

The various data pre-processing techniques employed in this study are presented in Table 1.

Table 1. Steps involved in data pre-processing task

Preprocessing Technique	Task Involved	Expected Output
Tokenization [16]	Text processing	List of individual words (tokens)
Lowercasing	Text normalization	Text with all letters converted to lowercase
Stopword Removal [17]	Text cleaning	Text with common stop words removed
Stemming [18]	Text normalization	Words reduced to their root or base form (stem)
Lemmatization [17]	Text normalization	Words transformed to their base or dictionary form (lemma)
Spell Checking [18]	Text cleaning	Corrected text with spelling errors addressed
Removal of Punctuation	Text cleaning	Text with punctuation marks removed
Handling Missing Data [18]	Data cleaning	Dataset with missing values imputed or removed
Scaling/Normalization	Feature scaling	Features scaled to a standard range (e.g., [0, 1])

Word Embedding of Student Responses

This phase of the framework actualizes feature extraction through word embedding functionality. The multi-layered mechanism known as a Bidirectional Encoder Representation from Transformer (BERT) is an encoder-only design that can be optimised for a range of downstream natural language processing applications, including question answering, named entity recognition, and text classification. BERT learns sentence-level coherence using a next-sentence prediction objective and bidirectional context from both left and right tokens using a masked language modelling objective while XLNET, an enhanced version of BERT, was used to benchmark for performance comparison [19]. The optimization process involves adjusting word vectors to improve the model's ability to predict content words accurately.

Topic Modeling of Student Responses

The LDA provides a probabilistic framework for uncovering latent topics in a collection of documents, making it a powerful tool for topic modeling in natural

language processing. Mathematically, the objective in LDA is to maximize the likelihood of observing the given set of documents D given the latent variables θ and β. This is typically done through variational inference or Gibb's sampling.

Random Forest Modeling

The Random Forest algorithm involves an ensemble of decision trees, where each tree is trained on a subset of the data and makes independent predictions. The final prediction is often obtained through a voting mechanism [20], and the algorithm provides a robust and interpretable framework for classification tasks. The specific steps for training and predicting with a Random Forest model include the construction of decision trees, feature selection at each split, and aggregation of predictions to produce the final output [21]. The details of these steps involve statistical measures like Gini impurity, which guide the tree-building process.

The Mathematical Representation of the Hybrid Solution

The mathematical formula Equation [4], and model for the methodology is presented in the following steps:

1. Initialization:

Let D be the set of documents.

Let *P*(T|D) be the topic distributions obtained from LDA.

Let *V* be the word vectors obtained from BERT.

Let *Y* be the target variable representing the grades.

2. Feature Concatenation:
    a. Concatenate the topic distributions *P*(T|D) and the word vectors *V* to create a hybrid feature set *X* for each document:

X = [P(T|D),V]

3. Random Forest Training:
    a. Train a Random Forest classifier using the hybrid feature set *X* and the target variable *Y*.
4. Random Forest Prediction:

Given a new document D_{new}, obtain its topic distribution *P*(*T*|D_{new}) and word vectors V_{new}

    a. Concatenate these features to create the hybrid feature set X_{new}= [*P*(*T*|D_{new}),V_{new}].
    b. Use the trained Random Forest model to predict the grade Y_{new} for the new document:

$$Y_{new} = RandomForestPredict(X_{new}) \quad (1)$$

The ensemble nature of Random Forest provides stability and generalization [20], making it a powerful algorithm for various machine-learning tasks.

Experimental Setting

The following segments were done for the complete implementation of the study:

Library Importation: The code imports necessary libraries for working with the data (extracted by BERT and LDA), machine learning, and GUI development using Tkinter. The key libraries used were pandas for data manipulation, sklearn for machine learning tasks, joblib for model saving and loading, and Tkinter for building the graphical user interface (GUI).

BERT hyperparameters: Selecting suitable hyperparameters that regulate the learning process and impact the model's performance is necessary for fine-tuning BERT. The learning rate, batch_size, max_seq_length, epoch, and other hyperparameters are the most important ones for a transformer-based model. The variables for a basic transformer are epoch = 40, max_seq_length = 50, batch_size = 24, and learning rate = 2e-4. It makes use of the AdamW optimiser by default and the dropout rate was set to 0. Mean pooling strategy was used for all the hidden layers.

The training part of the code involved loading a training dataset from a specified path on the computer (grading_aby_test.csv). The dataset is assumed to has 1000 attributes labeled n0 to n999 plus the word weights by LDA, and a target variable labeled 'grade', which represents the class (the score awarded by the teacher). The features (X_train) and labels (y_train) are then extracted. A Random Forest Classifier is instantiated and trained using the features and labels. The trained model was saved to a file (rf_model.joblib) for further use.

The testing part of the code involves loading a test dataset from a specified path on the computer (grading_aby_test.csv). The test dataset is assumed to have the same features as the training dataset, including the grade column. An attempt was made to match the features of the test dataset (test_data) with

the features of the training dataset (X_train). If a mismatch is detected, an error message is displayed in the lower space of the GUI. The 'grade' column is then dropped from the test dataset to ensure it matches the training data. The trained Random Forest model is loaded, and predictions are made on the test set. The predicted grades are saved to a CSV file (grading_mark.csv) and displayed in the lower space of the GUI.

SMART-CGS User's Interface was set up using Tkinter. It includes an upper space with a text box for displaying and uploading CSV file content, a button to upload a CSV file, a lower space with a text box for displaying predicted grades, and a button to initiate the prediction process. Users can upload a CSV file containing textual responses, and upon clicking the 'Grade' button, the predicted grades are displayed in the lower space.

Performance Evaluation

The hybridized models (BERT+LDA+RF and XLNET+LDA+RF) were evaluated using accuracy, precision, recall (sensitivity), and F1 Score for gauging the model's performance and effectiveness. The choice of the metrics and their methodological approach are presented in Table 2. Meanwhile, the engineering validation of the proposed system was done using the Inference Latency (IL) and scalability.

Table 2. Description of performance metrics for model evaluation

Metric Name	Meaning	Mathematical Formula
Accuracy	Proportion of correctly classified instances	$\frac{TP+TN}{TI}$
Precision	Proportion of true positives among predicted positives	$\frac{TP}{TP+FP}$
Recall (Sensitivity)	Proportion of true positives among actual positives	$\frac{TP}{TP+FN}$
F1 Score	The harmonic mean of precision and recall	$2 \times \frac{Precision \times Recall}{Precision + Recall}$
Inference latency	mean time taken to execute one student respond in millisecond (ms)	$IL = (EMT + CMT)ms$

Where TP = True Positive, TN = True Negatives, TI = Total instance, FP = False Positive, FN =False Negative, EMT = embedding mean time, and CMT = classification mean time.

4. Results

The implementation of the various phases of the research design was achieved and the results are presented in this section.

Data Acquisition

A graded examination result dataset was used for the implementation of this study. The lecturer examined twenty-eight (28) of his students and presented the question, responses, and grades accorded each of them. Table 3 shows the structure of an instance of the data (first eight instances), which is acquired for this study. The grading forms the class label of each response into the random forest algorithm.

Table 3. Instances of the acquired study data

QUESTION	What do you consider the relationship between Artificial Intelligence (AI) and the generative AI ChatGPT?	
Student_ID	Response	Grading
Pg_1	The symbiotic relationship between AI and ChatGPT fosters a dynamic interplay, where AI algorithms empower ChatGPT's language generation capabilities, while the contextual richness of ChatGPT enhances AI's ability to comprehend and respond effectively.	25
Pg_2	In exploring the intricacies of AI-ChatGPT collaboration, we observe a convergence of natural language understanding in AI and language generation in ChatGPT, shaping a synergistic alliance that propels advancements in human-computer interaction.	21
Pg_3	The recursive nature of AI leveraging ChatGPT and vice versa leads to a recursive amplification of language models, unveiling novel avenues for refining both AI's cognitive capacities and ChatGPT's contextual comprehension.	19
Pg_4	Investigating the reciprocal impact of AI algorithms on ChatGPT, we discern a balance, with AI fine-tuning ChatGPT's responsiveness, and ChatGPT contributing to AI's adaptability in processing diverse linguistic inputs.	22
Pg_5	The amalgamation of AI-driven methodologies and ChatGPT's natural language prowess creates a virtuous cycle, propelling the evolution of conversational agents by enhancing AI's analytical capabilities through contextualized language modeling.	16

Feature Extraction by BERT and LDA

Feature extraction was achieved on the responses of each student to output numeric data used to train the

random forest machine learning algorithm. Each of the student responses in CSV was inputted into the BERT algorithm using wordPiece library for the vertorization. The algorithm performs the word embedding to extract 768 feature vectors each. For all the responses, the extracted feature vectors were saved in a CSV file. The responses were also subjected to LDA. The topic modeling extracts the weights of the topical words contained in each of the responses. For the sample data earlier described in Table 3, some of the extracted topic weights by the LDA is presented in Table 4. The weights were added to their corresponding word embedding by BERT, and the concatenation forms the training set for the random forest modeling.

Table 4. Topics generated by LDA on the first student response

Topic No	Words in Topic	Numeric Weights
0	ai, ‘s, chatgpt, language, dynamic	0.050, 0.050, 0.050, 0.050, 0.050
1	chatgpt, ai, ‘s, language, effectively	0.110, 0.110, 0.076, 0.041, 0.041
2	interplay, contextual, effectively, relationship, dynamic	0.050, 0.050, 0.050, 0.050, 0.050
3	ai, chatgpt, ‘s, contextual, relationship	0.051, 0.051, 0.050, 0.050, 0.050
4	respond, dynamic, relationship, generation, interplay	0.050, 0.050, 0.050, 0.050, 0.050

Random Forest-based SMART-CGS

The two data containing the word embedding (by BERT) and the weights (generated by topic modeling) were concatenated to form the training set. All the attributes extracted by BERT and LDA were regarded as the independent variables while their corresponding grading (issued by the lecturer) was the dependent variable. The trained random forest algorithm did pattern recognition between the independent variables towards determining the dependent variable (grading) for each of the responses. The trained model (SMART-CGS) was then deployed for automatic grading. The feature has a plane where raw student responses were uploaded and displayed. This was achieved by the ‘Upload CSV File’ button. Upon clicking the button, a dialog box opens as in Fig. 1 that allows users to select the file where student responses were stored.

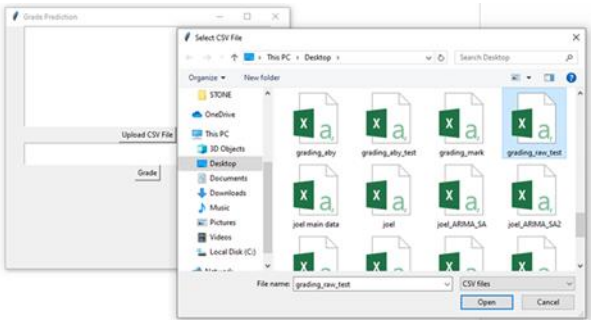

Fig. 1. The SMART-CGS prompting student responses’ selection

After the file selection, the student responses in the selected file were displayed on the GUI as seen in Fig. 2.

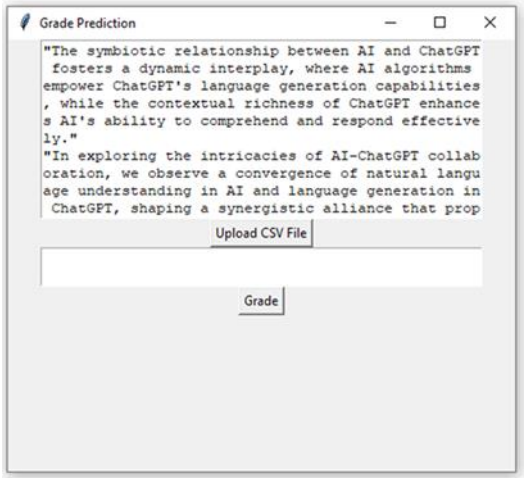

Fig. 2. GUI of the Random Forest-based SMART-CGS displaying the students’ responses

Upon the display of the student responses, the ‘Grade’ button activated the trained random forest model to automatically grade the responses and the student scores were stored in a CSV file displayed in Fig. 3 for ease of use by the lecturer.

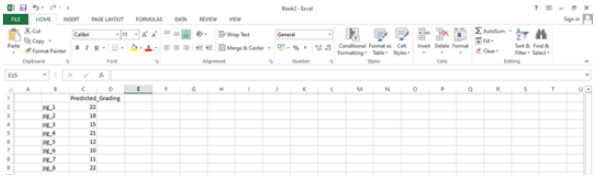

Fig. 3. Screenshot of the CSV file showing the Random Forest-based grading

Proposed Models Evaluation and System Validation

The performance evaluation result of the proposed models for SMART-CGS is presented in Fig. 4. The precision, recall, F1 score, and accuracy for BERT+LDA+RF and XLNET+LDA+RF are 0.92 and 0.94 respectively, and it indicate a high level of overall

models' performance. The BERT+LDA+RF model has a lower recall when compared with the precision and this implies that small portion of correct result has been compromised. Meanwhile, XLNET+LDA+RF outperformed its counterpart in all the outcome of the metrics and more importantly, there is a higher harmonic between the precision and recall value. This indicates that XLNET-based model was able to capture the semantic and syntactic meaning of the students' responses better and invariably gave a better grading prediction than the BERT-based model.

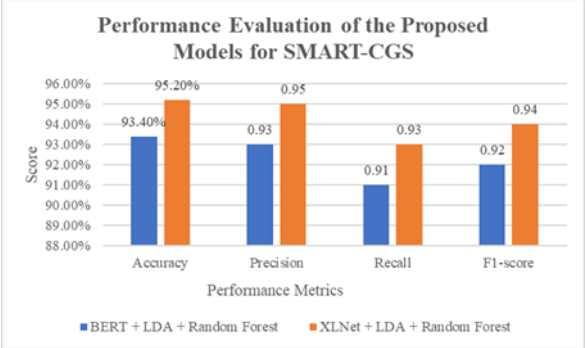

Fig. 4. Performance evaluation plot of the proposed models for SMART-CGS

The SMART-CGS system was validated using inference latency and scalability. Fig. 5 shows the graphical description of the inference latency for the models used. Inference latency calculate the mean time taken to execute one student respond in millisecond. From Figure 6, BERT-based model has a lower inference latency of 48.6ms as compared with XLNET-based model that has 76.5ms. Scalability, on the other hand, measures the throughput of the models to scale when tested with varying amount of student responses. In this study, the models were assessed with 100, 300, and 500 batch sizes of student responses. BERT-based model maintained a scaling of 47 seconds for completing 500 students' responses while XLNET-based model has 78 seconds for the same task.

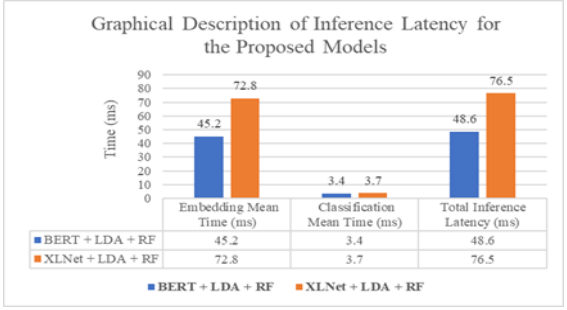

Fig. 5. Graphical Description of Inference Latency for the Proposed Models

5. Discussion

BERT-based model was chosen in this study because of its lightweight, scalability, low computation overhead, and efficiency. In this comparative analysis of predicted result using BERT-based model, the actual grading provided by the lecturer and the predicted grading generated by the SMART-CGS for each undergraduate student is examined. Fig. 6 shows the comparison between the actual grade and the predicted grades. The RF model's predictions generally show a trend of underestimating the actual grades, suggesting a conservative approach in its evaluations. There is a degree of alignment between the actual and predicted grades, it is evident that the SMART-CGS tends to assign lower scores because of slight variation in recall and precision values. Notably, for undergraduate students pg_5, pg_6, and pg_7, the model's predictions deviate significantly from the actual grades, indicating a potential limitation in capturing the expressions and depth of understanding demonstrated by these students, as opined by their lecturer. Additionally, the model struggles to accurately predict the higher grades, as observed in the case of pg_2 and pg_8, where the predicted grades fall short of the actual grades. This suggests a potential area for improvement in the model's sensitivity to sophisticated expressions, reinforcing the importance of continuous refinement and validation in leveraging an enhanced version of BERT such as XLNET embedding for complex extraction of semantic dependences among the sentences in grading tasks.

Although, BERT+LDA+RF has displayed a significant high performance across the metrics (0.92) but XLNET+LDA+RF performed better (0.94). The observed lower grading between the actual and predicted grades, considering the background of using BERT for vector extraction and LDA for topic weight determination as independent variables for training the Random Forest (RF) model, sheds light on the limitations of feature representation in the context of grading undergraduate responses. BERT, employed for embedding responses into vector representations, captures semantic relationships among words but might struggle to capture the intricacies of the expressions and depth of understanding in longer sentences or complex language structures. On the other hand, LDA, which assigns weights to topics within responses, provides a higher-level thematic overview

but may miss the finer details. The combined use of these features in the RF model seems to result in a conservative grading pattern, potentially emphasizing the broader themes at the expense of the expressions. Meanwhile, inference latency and scalability results are 48.6ms, 47s, and 76.5ms, 78s for BERT-based and XLNET-based models respectively. The faster throughput time for BERT indicates a faster processing speed and high scalability for a real-time content grading that can be integrated into school portals whereas, XLNET is useful when complex semantic intricacies need to be uncovered. More so, XLNET is quite expensive because of high computation overhead and it is slow in processing task.

Finally, the results of this work show a distinct trade-off between speed and accuracy between the two models implemented. BERT-based model is lightweight and effective which makes it recommendable to institutions having numerous students' responses to be graded daily while XLNET-based model is computationally heavy but has complex sematic understanding of sentences for higher grading accuracy.

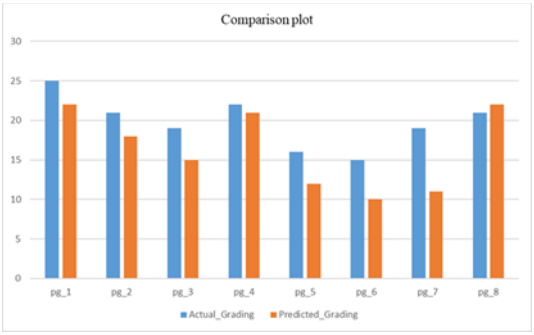

Fig. 6. Comparative analysis of the actual and predicted grading

Conclusion

This study aimed to pioneer a hybrid grading system named "SMART-CGS" for undergraduate students' assessments, leveraging the capabilities of machine learning and Natural Language Processing (NLP). The specific objectives were set to comprehensively review existing literature on automated grading systems within the machine learning and NLP context, followed by the development and implementation of a novel hybrid grading model. This model integrated topic-based features extracted through Latent Dirichlet Allocation (LDA) and word embedding-based features obtained using BERT and XLNET. The performance of this hybrid system was rigorously evaluated using standard machine learning metrics. The study has successfully contributed to the growing body of knowledge in automated grading systems by introducing a novel hybrid approach that incorporates both thematic understanding and semantic relationships within responses. The use of BERT and LDA for feature extraction demonstrated the potential for capturing the diverse aspects of undergraduate responses. The SMART-CGS trained on these features, exhibited promising performance metrics during training, with precision, recall, F1 score, and accuracy all reaching approximately 0.92. However, while the results are encouraging, it is essential to acknowledge certain considerations. The observed conservative grading pattern, potentially influenced by the dominance of lower grades in the dataset, suggests the need for ongoing refinement in feature engineering which prompted the usage of XLNET that gives a better result (0.94). The choice of the model to adopt depends on the speed and accuracy trade-off that existed between the two models.

References

[1] Haller, S., Aldea, A., Seifert, C. and Strisciuglio, N. "Survey on Automated Short Answer Grading with Deep Learning: From Word Embeddings to Transformers", *arXiv* 2022, arXiv:2204.03503

[2] Zhou, N. "Evaluating Human and Machine Assessment: Introducing a Hybrid Approach for Enhanced Educational Evaluation", *5th International Conference on Education Innovation and Philosophical Inquiries*, pp. 118-124, 2024 https://doi.10.54254/2753-7048/58/20241716

[3] Xu, Y., and Brown, G. T. L. "Teacher assessment literacy in practice: A reconceptualization", *Teaching and Teacher Education*, vol. 58, 149–162, 2016. https://doi.org/10.1016/j.tate.2016.05.010

[4] Gawand, D. V., Badi, M. H., Makharoumi, M. K. and Cain, M. R., "Study Design and Implementation of NLP Techniques for Automated," *International Journal of Innovation in Computational Science and Engineering,* vol. 2, no. 1, pp. 1-8, 2021.

[5] Kucak, D., Juricic, V. and Dambic, G. "Machine Learning in Education - a Survey of Current Research Trend", *Annals of DAAAM & Proceedings*, 29, 1726-9679, 2018, Vienna, Austria https://doi.10.2507/29th.daaam.proceedings.059

[6] Mpu, Y. "Bridging the Knowledge Gap on Special Needs Learner Support: The Use of

Artificial Intelligence (AI) to Combat Digital Divide Post-COVID-19 Pandemic and beyond–A Comprehensive Literature Review", *In: Intellectual and Learning Disabilities - Inclusiveness and Contemporary Teaching Environments*, 2023, edited by Altinay, F. and Altinay, Z. https://doi.10.5772/intechopen.113054

[7] Chen, X., Xie, H., Zou, D., & Hwang, G.-J. "Application and Theory Gaps during the Rise of Artificial Intelligence in Education", *Computers and Education: Artificial Intelligence,* vol. 1, 2020, Article ID: 100002. https://doi.org/10.1016/j.caeai.2020.100002

[8] Olaleye, T. O., Ugege, P., Okewale, O., Akinade, O., Akintunde, O. and Akparanta, C., "Evaluation of Vader and MultiLingual sentiment analyzers for opinion analytics using graphical illustrations," *in Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2022).,* Gliwice, Poland, 2022.

[9] Kulshretha, S. and Lodha, L., "Performance Evaluation of Word Embedding Algorithms," *International Journal of Innovative Science and Research Technology,* vol. 8, no. 12, pp. 1555-1561, December 2023.

[10] Kuyoro, S. O., Eluwa, J. M., Akinsola, J. E., Ayankoya, F. Y., Omotunde, A. A. and Adegbenjo, A. A., "Intelligent Essay Grading System using Hybrid Text Processing Techniques," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 6, no. 5, pp. 264-270., 2020.

[11] Rambola, R.K., Bansal, A., Savaliya, P., Sharma, V., Joshi, S. "Development of Novel Evaluating Practices for Subjective Answers Using Natural Language Processing", in Singh Pundir, A.K., Yadav, A., Das, S. (eds) *Recent Trends in Communication and Intelligent Systems. Algorithms for Intelligent Systems, 2021,* Springer, Singapore. https://doi.org/10.1007/978-981-16-0167-5_21

[12] Wang, X., Zhang, D., Asthana, A., Asthana, S., Khanna, S. and Verma, C. "Design of English hierarchical online test system based on machine learning" *Journal of Intelligent Systems,* vol. 30, no. 1, 2021, pp. 793-807. https://doi.org/10.1515/jisys-2020-0150

[13] Yuan, S., He, T., Huang, H., Hou, R., Wang, M. "Automated Chinese Essay Scoring Based on Deep Learning", *Computers, Materials & Continua*, vol.65, no. 1, 817–833, 2020. https://doi.org/10.32604/cmc.2020.010471

[14] Bernard, J., Sonnadara, R., Saraco, A. N., Mitchell, J. P., Bak, A. B., Bayer, I. and Wainman, B. C. "Automated grading of anatomical objective structured practical examinations using decision trees: An artificial intelligence approach", *Anat Sci Educ.* Vol.17, no. 5, 967-978, 2024. https://doi.10.1002/ase.2305

[15] Azeta, A., Guembe, B., Ankome, T. and Osakwe, J. "Machine Learning Techniques for Automatic Long Text Examination in Open and Distance Learning", *in Proceedings of the International Conference on Information Systems and Emerging Technologies (ICISET)*, 2020. http://dx.doi.org/10.2139/ssrn.4331526

[16] X. Sun, X. Liu, J. Hu and J.Zhu, "Empirical studies on the nlp techniques for source code data preprocessing," *in Proceedings of the 2014 3rd international workshop on evidential assessment of software technologies*, 2014.

[17] A. Petukhova and N. Fachada, "TextCL: A Python package for NLP preprocessing tasks," *SoftwareX,* vol. 19, p. 101122, 2022.

[18] A. Tabassum and R. R. Patil, "A survey on text pre-processing & feature extraction techniques in natural language processing," *International Research Journal of Engineering and Technology,* vol. 7, no. 6, pp. 4864-4867, 2020.

[19] Aurpa, T. T., Fariha, K. N., Hossain, K., Jeba, S. M., Ahmed, M. S., Adib, M. R. S., Islam, F. and Akter, F. "Deep transformer-based architecture for the recognition of mathematical equations from real-world math problems," *Heliyon,* vol. 10, no. 20, e39089. https://doi.org/10.1016/j.heliyon.2024.e39089

[20] Schonlau, M. and Zou, R. Y. , "The random forest algorithm for statistical learning," *The Stata Journal,* vol. 20, no. 1, pp. 3-29, 2020.

[21] Charbuty, B. and Abdulazeez, A., "Classification based on decision tree algorithm for machine learning," *Journal of Applied Science and Technology Trends,* vol. 2, no. 01, pp. 20-28, 2021.