WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Enhancing Education with NLP: Automated Grading and Feedback System

Kondavalasa Jyothsna[1], Meesala Sravani[2]

[1]Department of Computer Science and Engineering, GMR Institute of Technology, Rajam, 532127, India.
[2] Department of Computer Science and Engineering, GMR Institute of Technology, Rajam, 532127, India.

ABSTRACT :

The expansion of educational technology was pretty quick, with the need for automatic grading and feedback systems to keep pace in order to maintain scalability and efficiency in evaluating students. These are crucial for mainly the scale involved-cumbersome in the event of manual grading. Where previous approaches have failed to guarantee any meaningful result lies in their use of NLP models including traditional RNNs, LSTMs, and even early transformer models like BERT, employed in tandem with machine learning techniques such as Random Forest and K-Means Clustering, to automate grading student responses. But these models often have trouble arriving at a degree of precision and personalization needed to provide meaningful feedback on many diverse, nuanced submissions by the students. Introduction of the methodology on a new approach using Sentence-BERT with the aspect to increase precision and personalization in automated feedback is one of the focus points of this review. The approach fits the nuances in patterns of student responses better than earlier models, which would make the impact of the feedback more relevant for each learner's learning needs. Such computational methods provide for a resilient and effective approach to a scaleable combination for grading with a new standard set toward using automated educational tools.

KEYWORDS:- Automatic Grading, Feedback System, Natural Language Processing, Transformer Models, BERT, Sentence-BERT

1. INTRODUCTION

In this ever-changing technology world, the call for efficient but scalable evaluation systems has expanded. Large numbers of students and diversified learning environments make manual assessment both time-consuming and inefficient in this large educational setup. For these reasons, automated assessment systems have been developed. These systems deal with how one can reduce the burden on educators while ensuring the quality and relevance of responses given to the learners.

Traditional automated grading systems have relied on NLP models including Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks and even early transformer architectures like Bidirectional Encoder Representations from Transformers (BERT). Such models combined with machine learning techniques such as Random Forest and K-Means clustering resulted in promising initial results. But more often than not, these approaches lack the accuracy level and response type required for a meaningful evaluation of students. They fail to have a sensitive ability to reflect subtle nuances between diverse types of responses from students. Thus sometimes they are general, imprecise, and not tailored to proper learning needs.

A new approach will be proposed with the use of a transformer-based model, designed to embed sentences, called Sentence-BERT to enhance accuracy and customization of automated responses. Overcoming the problems of the earlier models, Sentence-BERT provides better representation of students' responses, capturing fine-grained semantic differences and adding significant nuance to context-aware responses. With this method, automation assessment systems can give better and more relevant and tailored responses to ensure that learners get a better learning experience.

Through such a methodology that introduces Sentence BERT into the framework of an automated assessment, we propose to enhance the accuracy of the evaluation of responses beyond what can be done by using alignment with the pedagogical goal of personalized learning. This method provides a benchmark toward creating reliable educational resources with high efficiency for transformation in automated responses across diverse educational environments.

2. Related work

Research in student performance predictors and automatic grading systems is numerous. There is Kenton et al. (2019) [4] who developed the BERT model, revolutionizing the understanding of the natural language with the implementation of the use of bidirectional transformers to tremendously improve the performance in a wide range of tasks, including Automatic Short-Answer Grading (ASAG), improving sentence-level understanding.

Zhu et al. in 2022 [1] extended the BERT-based approach specifically designed for ASAG by incorporating Bi-LSTM and Capsule networks. Capturing both global and local semantic contexts gave more accuracy in grading outcomes across multiple datasets. Such a system inherently presents a considerable improvement in the automatic grading of short answers.

Another work in this direction was the design of a multiclass prediction model based on machine learning presented by Bujang et al. (2021) [7] to predict student grades. Their system overcomes the problem of imbalanced datasets with the SMOTE technique and added the feature selection method to increase the accuracy of the prediction and data visualization assistance to observe the performance of students over time. Sung et al. (2019) [11] tuned BERT on short-answer grading tasks, mainly concerning cross-domain generalization. They presented a model that adapts to multiple domains-thus proving that transformer-based models can be used effectively in grading even in highly diversified educational settings.

Shiva Shankar and Ravibabu, (2019) [6] of making use of NLP as an approach for grading digital reports. Their approach emphasizes key NLP techniques such as tokenization and feature selection that, in turn, gives more consistent grading by emphasizing relevant features of the student essays, such as word frequency and sentence structure.

Developed by Del Gobbo et al. in 2023 [18], GradeAid is a framework for automatic grading of short answers using lexical and semantic analysis. It supports non-English datasets, thus expanding its application to different educational contexts and improving grading accuracy through robust validation techniques.

Liu et al. (2019) [16] focused on programming assignment grading by introducing AUTOGRADER, a formal-semantic-based tool. This system compares student submissions to reference implementations, offering immediate feedback to students on their programming errors, which accelerates their learning process.

Xue et al. (2021) [2] also have another approach, multi-dimensional essay scoring, using a hierarchical BERT-based transfer learning method. It segmented longer texts into several sections to ensure that key information is retained and also gave detailed feedback on varying dimensions of writing quality. By using this approach, this model has demonstrated its ability to generalize across different essay topics.

Ahmed et al. (2023) [20] attempted the task using models such as Random Forest, Naive Bayes, and LSTM for sentiment classification of student feedback. Their research was a reflection of how well LSTM-based models compare in capability to other models in capturing students' sentiment-adding the profound insights to enhance educational experience.

In this regard, Sahu and Bhowmick 2019 [3] focused upon feature engineering, utilizing several regression models in an ensemble-based approach to enhance ASAG. Upon integrating relevance feedback with the topic modeling of responses, it was possible to significantly boost the accuracy of the system while also making it more resilient against diverse response patterns from students.

Similarly, Süzen et al. (2020) [5] devised an approach that automatically evaluates small answer questions based on the similarity of words computed between a student's response and a model answer. It enhanced both the effectiveness and efficiency in grading large volumes of student work and also delivers detailed feedback on commonly occurring mistakes.

Osakwe et al. (2023) [17] applied RL in optimizing self-regulated learning strategies. The authors demonstrated how significant RL could enhance student learning outcomes and performance via comparison with LSTM and genetic algorithms.

Chen and Ward (2021) [9] was a more niche contribution since it targeted the performance prediction of programming classes. They derived models of decision trees and linear regression models for the auto-grading system data to predict the final exam score of students based on their submission behavior and passing rate.

Lu and Cutumisu introduced a system generating feedback using deep learning models, namely CNN and LSTM, in 2021 [13]. Their system uses Constrained Metropolis-Hastings Sampling to generate relevant feedback. This system enhances essay scoring with feedback that is not only accurate but also semantically aligned with content.

Hellman et al. (2020) [14] examined the potential application of MIL to predict where feedback should be localized in an essay. His model is sentence-level, which means his model can give very detailed feedback without having to use annotated data, a feature that makes it extremely sensitive to large educational environments.

Chauhan et al. (2020) [12] developed an automated grading system for the theoretical content using models including Random Forest and Word2Vec. This system processes theoretical answers by extracting the basic textual features, thereby providing consistent and reliable grading of technical subjects.

Sadanand et al. (2022) [15] proposed an AES using LSTM models; the system was combined with sentiment analysis in order to align students' responses with expected feedback. The output of the system is that students can gain in-depth, personalized feedback so as to improve their writing skills.

Bernius et al. (2022) [10] developed the CoFee system for massive open online courses. Automation of textual student answers' feedback allowed the system to group student response segments into similar groups, hence automatic and reusable with precision and saving instructors time.

Guo et al. (2019) [19] presented a self-adaptive classifier ensemble model, that was developed for credit scoring but can also be applied in ASAG. Based on the multi-layer stacking over multiple classifiers, they managed to achieve very high accuracy in the prediction performance.

Zhang et al. (2022) [8] report a grade semi-open-ended questions based on LSTM networks. Their knowledge domain-based system merges two types of knowledge: domain-specific and general, and can therefore grade in several languages while ensuring accurate grading.

3. METHODOLOGY

3.1 Data Collection:-

Every automated assessment and response system depends highly on the qualities and diversity of the dataset they rely on to train, validate, and test their algorithms. This study includes a well-curated dataset of student short answer responses to an environmental science assessment. Below is a data collection process, pre-processing steps, and features used for this particular study.

Source: The dataset is students' answers collected from actual educational platforms on the internet. Those include online quizzes, assignments, and tests. They span several topics, ranging from literature to science and social studies, ensuring different types of questions and complexities.

Grading Annotations: Responses elicited from students have been graded manually by human graders. Human-graded responses are used as ground truth to train the system. In addition to grades, the data set contains in detail answers provided by human graders, which is used as reference for response generation done by the automated system.

Response Diversity: The responses are of students with varying skill levels. This ensures the model generalizes well across skill levels and backgrounds. It covers all potential responses from the correct ones to partly correct to incorrect, in an attempt to capture fine-grained differences in quality.

3.2 Data Preprocessing:-

Tokenization

Tokenization is the process of breaking down a text into tokens. According to the type of tokenizer, units could be words, characters or subwords. Essentially, the aim of this is to transform a text document into a list of more manageable smaller parts, for example, words or subwords. Word tokenization breaks texts into words by spaces or marks of punctuation. Subword tokenization, on the other hand, refers to a process when words that are not found in the vocabulary of a model need to be broken into smaller known units of subword components. There are numerous libraries used in performing different tasks on tokens. NLTK provides very simple word-level tokenization. High-performance is provided by SpaCy, and Hugging Face's Transformers library support advance versions including subword techniques.

Noise Removal

Noise removal is a preprocessing step in NLP; it just filters out the unnecessary elements or the extraneous elements present in the text. Generally, it's known as "noise," such as punctuation marks, special characters, HTML tags, or simply irrelevance tokens that add nothing to the meaningful analysis of the text. Removing +this noise from the responses of students makes the input cleaner and more structured. This is one process very essential in enhancing the embedding generation during the model performance improvement. This process means that the model is focused on the meaningful content of the text and not sidetracked by unnecessary or irrelevant things.

Lemmatization

Lemmatization is the process of rewriting words into their base or root forms, referred to as lemmas. A lemmatizer used is such a tool as WordNet or spaCy. Converting words to root form can help in the normalizing of text so that different forms of the same word that contain the same lemma, for example "running" and "run," are considered identical. This is important because it minimizes the variability in the text, where semantically similar words come to be grouped into the same cluster, improving the quality and effectiveness of embeddings generated by machine learning models. That is, lemmatization can help simplify and standardize language to let models more easily understand and process.

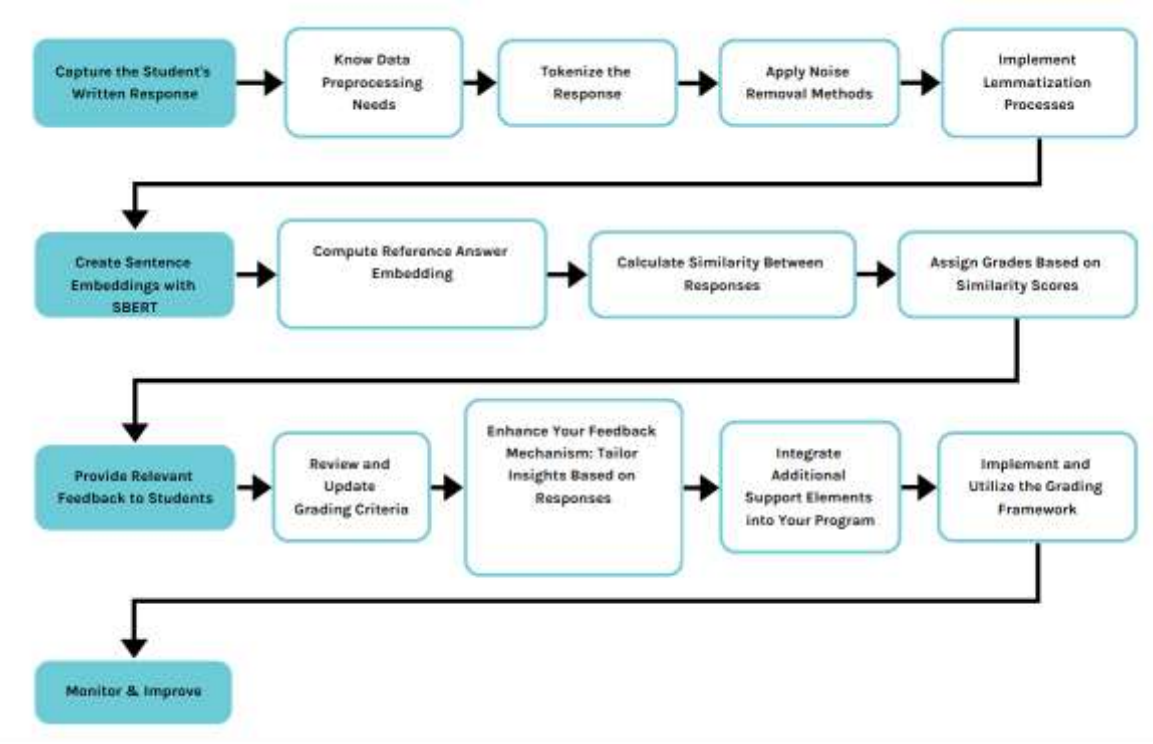

Fig. 1 – Methodology for calculating grading and feedback

3.3 Sentence-BERT Model Architecture :

It is the modified BERT, which stands for Bidirectional Encoder Transformer. It is a variant of the Transformers model aimed at generating representations. Sentence embeddings can help determine how closely two meanings are related. This section will delve deep into SBERT architecture to understand how it varies from the BERT and how it may be utilized in systems for automated assessments and responses.

Overview of BERT Architecture

BERT is a transformer-based model that encodes input text using a deep, bidirectional architecture. The model learns contextual representations of words by taking into account the entire sentence both from the left and the right- with a mechanism called "masked language modeling." Although BERT can capture deep context-based data, it was not designed for efficient sentence-level similarity tasks. The traditional BERT models are often too computationally expensive to compare multiple sentences directly, as they are designed to process sentence pairs jointly, resulting in significant computation overhead.

Key Features of BERT:

Bidirectional Encoder: BERT reads text from both ends to get full contextual understanding.

Transformer Network: BERT is developed based on the Transformer architecture, which uses attention mechanisms to focus different parts of the input to understand properly.

Masked Language Model (MLM): When training, BERT masks random words in the sentence and predicts the masked words based on the context of the other words.

Next Sentence Prediction, or NSP: BERT is also trained to predict when two sentences could be consecutive in text; this helps with sentence relationship tasks.

While the standard BERT model has high contextual understanding, it computes embeddings for each possible pair of sentences when in similarity comparison, which is computationally expensive and not practical for huge comparisons between sentences like those used in automated assessment systems.

Modifications in SBERT

SBERT modifies BERT's architecture to create more efficient and scalable sentence-level embeddings for tasks like automated assessment and responses generation:

Independent Sentence Embeddings: Instead of processing two sentences together, SBERT generates embeddings for each sentence separately. This allows for quick and efficient comparison of student responses and model answers.

Siamese Network Structure: SBERT uses a Siamese network where two identical BERT models take two input sentences separately to produce the embeddings. These embeddings can be compared by similarity measures such as cosine similarity to answer tasks like assessment**.**

Pooling Layer: After the BERT model processes each sentence, SBERT adds a pooling layer to convert the output into a fixed-length vector representation (embedding). This vector summarizes the entire sentence's meaning, which is crucial for assessment and responses analysis.

Sentence-BERT (SBERT) Architecture Workflow:

The SBERT workflow is a step-by-step process describing input sentences being taken through the model to create meaningful sentence embeddings and apply these embeddings to tasks such as assessment and responses analysis. Below is an explanation of how the entire SBERT architecture workflow develops from input tokenization to similarity comparison.

Encoding via BERT Layers

After tokenization, the input sentence goes through multiple transformer layers in the pre-trained BERT model. It is intended to capture relationships for any word in the sentence. Some of the key points are:

Transformer Architecture: SBERT leverages BERT's multi-layer transformer architecture, where each layer applies self-attention to model the dependencies between words, regardless of their position in the sentence.

Contextualized Embeddings: The output of each transformer layer is a contextual embedding for every token in the sentence. Using vectors, each word encapsulates its meaning relative to the other words in the sentence.

For example, in the sentence, "The sky is blue," the word "blue" will hold a different meaning than if the sentence was "The ocean is blue," as it dependent upon context.

Pooling Layer for Sentence Embeddings

The pooling layer takes the sequence of embeddings of tokens and transforms it into a fixed-length sentence embedding. In SBERT, a pooling strategy is used to summarize all the information contained in all tokens into one vector that represents the whole sentence. Common Pooling Strategies:

1. Mean Pooling:

This is the most common approach in SBERT. It finds the mean pooling, that computes the average of all token embeddings, for example, "The", "sky", "is", and "blue". The resulting embedding then becomes a 768-dimensional vector, in case using BERT-base, which represents the whole sentence. Formula for mean pooling:

Sentence_embedding = (E1 + E2 + ... + En) / n

where `E1, E2, ..., En` are the embeddings for each token in the sentence, and `n` is the number of tokens.

2.[CLS] Token Pooling:

In this method, the `[CLS]` token embedding, which is generated at the start of the sentence, is used as the final sentence embedding.The `[CLS]` token is designed to capture the global meaning of the entire sentence.

3. Max Pooling:

For every dimension of the token embeddings, max pooling selects a maximum value across all token embeddings, resulting in an embedding for the sentence which gives a substantial emphasis on the most striking features of the sentence. Every one of these methods can be used based on the task. However, mean pooling is largely preferred in SBERT as it tends to remove the contribution of each word and has a better well-balanced representation of the sentence.

Embedding Generation:

After pooling is conducted, SBERT generates a fixed-size embedding vector for every sentence. Embedding is a dense representation of the semantic content of the sentence. Typically, this embedding vector is 768 dimensions in size if the BERT-base model is used (or smaller dimensions if using lighter versions like DistilBERT or MiniLM). Some important points of embedding generation are

Semantic Encoding: This results in an embedding that is a dense vector; each dimension captures something about the meaning of the sentence. Such embeddings are designed to be maximally informative about the semantics of the sentence. The objective of an embedding is always optimized towards tasks that resemble sentence similarity, ranking, or clustering.

Efficient Representation: Since SBERT processes each sentence independently, these embeddings can be precomputed and stored, allowing for quick and efficient comparisons later.

3.4 Sentence Similarity Calculation :

Once the sentence embeddings are generated, the next step is to calculate the similarity between them. In the context of assessment and responses analysis, the student's response and reference answer embeddings are compared to assess how similar they are in meaning.

Common Similarity Metrics are :

1. Cosine Similarity:

This is the most commonly used similarity metric in SBERT. Cosine similarity measures the angle between two embedding vectors in high-dimensional space. A cosine similarity score near 1 depicts that the given response by the student and the reference answer exhibit semantic similarity whereas a score near 0 depicts that the given response by the student and the reference answer contain low semantic similarity.

Formula for cosine similarity:

cosine_similarity(A, B) = (A • B) / (|A| * |B|)

where `A` and `B` are the embedding vectors of the student response and the reference answer, and `•` denotes the dot product between them.

2. Euclidean Distance:

Another way to measure similarity is by calculating the Euclidean distance between the two sentence embeddings.Smaller Euclidean distances indicate higher similarity, whereas larger distances indicate dissimilarity.

3.5 Grading and Feedback Generation :

Once the similarity score of the student's answer versus the reference answer is computed, it can then be exploited for numerous downstream tasks:

Grading:

The grade can be obtained directly from the similarity score. For consistency in the grades allocated, the grading may be classified into different tiers against set levels. Additionally, the score may be adjusted to include partial credit as a factor, giving greater weight to nuances of the students' responses. For example:

A cosine similarity score of 0.9–1.0 might correspond to a high grade (90-100%).

A score of 0.7–0.9 might correspond to a medium grade (70-90%).

A score below 0.7 might correspond to a lower grade (less than 70%).

Feedback Generation:

Predefined Feedback Templates: Based on the similarity score, SBERT can trigger predefined responses templates.

For example:

If the score is high: "Excellent! Your answer closely matches the expected response."

If the score is medium: "Good attempt, but your answer lacks some key details."

If the score is low: "Your answer is missing key concepts. Please review the material."

Personalized Feedback: SBERT can also be used to provide personalized responses based on multiple student responses by computing the semantic differences of those responses. It may identify gaps in students' knowledge and thus suggest relevant resources.

4.CONCLUSION:

The work introduced a new methodology for assessing responses with SBERT, which addresses all the critical challenges in accuracy and the customisation problems of educational technology. Our approach captures the semantic nuances of student responses effectively to highly improve the accuracy of the automated assessment compared to state-of-the-art traditional models like LSTMs and RNNs. In fact, the integration of K-Means Clustering integrates the actual recognition of student answer patterns. In that way, it is possible for a better response to face an individual need in learning. In results, there is an obviously increased relevance and personalization that will surely set new standards for automated educational tools.

On the one hand, this research contributes to the advancement of scalable assessment systems and further supports the necessity of meaningful response provision in large-scale educational environments. Finally, the present work provides a pretty solid basis for the development of versatile, robust automated assessment systems that can finally correspond to the needs of today's diversified student populations and open up more effective and personalized opportunities for learning and assessment.

REFERENCES:

[1] Xue, J., Tang, X., & Zheng, L. (2021). A hierarchical BERT-based transfer learning approach for multi-dimensional essay scoring. Ieee Access, 9, 125403-125415.

[2] Sahu, A., & Bhowmick, P. K. (2019). Feature engineering and ensemble-based approach for improving automatic short-answer assessment performance. IEEE Transactions on Learning Technologies, 13(1), 77-90.

[3] Kenton, J. D. M. W. C., & Toutanova, L. K. (2019, June). Bert: Pre-training of deep bidirectional neural transformer networks for language understanding. In Proceedings of naacL HLT (Vol. 1, p. 2).

[4] Süzen, N., Gorban, A. N., Levesley, J., & Mirkes, E. M. (2020). Automatic short answer assessment and responses using text mining methods. Procedia computer science, 169, 726 743.

[5] Shiva Shankar, R., & Ravibabu, D. (2019). Digital report assessment using NLP feature selection. In Soft Computing in Data Analytics: Proceedings of International Conference on SCDA 2018 (pp. 615-623). Springer Singapore.

[6] Bujang, S. D. A., Selamat, A., Ibrahim, R., Krejcar, O., Herrera-Viedma, E., Fujita, H., & Ghani, N. A. M. (2021). Multiclass prediction model for student grade prediction using machine learning. Ieee Access, 9, 95608-95621.

[7] Zhang, L., Huang, Y., Yang, X., Yu, S., & Zhuang, F. (2022). An automatic short answer assessment model for semi-open-ended questions. Interactive learning environments, 30(1), 177-190.

[8] Chen, H., & Ward, P. A. (2021). Predicting student performance using data from an auto-assessment system. arXiv preprint arXiv:2102.01270.

[9] Bernius, J. P., Krusche, S., & Bruegge, B. (2022). Machine learning based responses on textual student answers in large courses. Computers and Education: Artificial Intelligence, 3, 100081.

[10] Sung, C., Dhamecha, T. I., & Mukhi, N. (2019). Improving short answer assessment using transformer-based pre-training. In Artificial Intelligence in Education: 20th International Conference, AIED 2019, Chicago, IL, USA, June 25-29, 2019, Proceedings, Part I 20 (pp. 469-481). Springer International Publishing.

[11] Chauhan, R. K., Saharan, R., Singh, S., & Sharma, P. (2020). Automated content assessment using machine learning. arXiv preprint arXiv:2004.04300.

[12] Lu, C., & Cutumisu, M. (2021). Integrating Deep Learning into an Automated Feedback Generation System for Automated Essay Scoring. International Educational Data Mining Society.

[13] Hellman, S., Murray, W. R., Wiemerslage, A., Rosenstein, M., Foltz, P., Becker, L., & Derr, M. (2020, July). Multiple instance learning for content responses localization without annotation. In Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications (pp. 30-40).

[14] Sadanand, V. S., Guruvyas, K. R. R., Patil, P. P., Acharya, J. J., & Suryakanth, S. G. (2022). An automated essay evaluation system using natural language processing and sentiment analysis. International Journal of Electrical and Computer Engineering, 12(6), 6585-6593.

[15] Liu, X., Wang, S., Wang, P., & Wu, D. (2019, May). Automatic assessment of programming assignments: an approach based on formal semantics. In 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET) (pp. 126-137). IEEE.

[16] Osakwe, I., Chen, G., Fan, Y., Rakovic, M., Li, X., Singh, S., ... & Gašević, D. (2023). Reinforcement learning for automatic detection of effective strategies for self-regulated learning. Computers and Education: Artificial Intelligence, 5, 100181.

[17] Osakwe, I., Chen, G., Fan, Y., Rakovic, M., Li, X., Singh, S., ... & Gašević, D. (2023). Reinforcement learning for automatic detection of effective strategies for self-regulated learning. Computers and Education: Artificial Intelligence, 5, 100181.

[18] Guo, S., He, H., & Huang, X. (2019). A multi-stage self-adaptive classifier ensemble model with application in credit scoring. IEEE Access, 7, 78549-78559.

[19] Ahmed, N., Khouro, M. A., Dawood, M., Dootio, M. A., & Jan, N. U. (2023). Student textual responses sentiment analysis using machine learning techniques to improve the quality of education. Pakistan Journal of Engineering, Technology & Science, 11(2), 32 40.

[20] Mizumoto, A., & Eguchi, M. (2023). Exploring the potential of using an AI language model for automated essay scoring. Research Methods in Applied Linguistics, 2(2), 100050.