2024 8th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS) | 979-8-3315-0546-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/CSITSS64042.2024.10817016

# Leveraging LLM and RAG for Automated Answer Script Evaluation

Pranav Darshan
*Department of Computer Science and Engineering*
*R V College of Engineering*
Bengaluru, India
pranav_darshan@outlook.com
https://orcid.org/0009-0004-5586-3994

Nihar Mandahas
*Department of Computer Science and Engineering*
*R V College of Engineering*
Bengaluru, India
niharmandahas@gmail.com
https://orcid.org/0009-0009-5313-2094

Pratheek Rao MP
*Department of Computer Science and Engineering*
*R V College of Engineering*
Bengaluru, India
mppratheek@gmail.com

Raghuveer Narayanan Rajesh
*Department of Computer Science and Engineering*
*R V College of Engineering*
Bengaluru, India
raghuveer0508@gmail.com
https://orcid.org/0009-0002-8511-4260

Deepamala N
Associate Professor
*Department of Computer Science and Engineering*
*R V College of Engineering*
*Bengaluru, India*
deepamalan@rvce.edu.in
https://orcid.org/0000-0001-8594-2248

***Abstract*—In the realm of education, examinations play a very significant role. If writing an exam is considered tedious, the process of evaluating hundreds of answer scripts can be even more daunting. It usually takes weeks to evaluate all the answer scripts, and there is always a factor of bias in human corrections. Hence, to eliminate all these problems, this study uses a large language model to automate this task and eliminate human bias. The study leverages a fine-tuned LLM to assess answer scripts related to the subject of operating systems using a specific dataset. Along with the LLM, it incorporates Retrieval-Augmented Generation (RAG) to get the context of a given question from a prescribed textbook. The proposed platform also has the capability of analyzing handwriting from an actual answer script, and then this is passed onto the model as input along with context from RAG. Finally, the entire system is integrated into an interactive web platform, deployed using AWS SageMaker. By combining all the technology mentioned, this study has made a sincere attempt to solve the burden of correcting manuscripts.**

**Keywords—AWS SageMaker, fine tuning, large language models (LLMs), Retrieval Augmented Generation (RAG)**

## I. Introduction

Assessing handwritten responses can take a lot of time and can be a tedious task for educators, which is a diversion from research and student engagements. The manual evaluation can also involve bias, leading to unfair assessments. To address these challenges, an automated solution has been developed that leverages machine learning.

The innovative workflow aims to streamline the assessment process, reduce the workload on professors, and ensure more consistent and unbiased evaluation. The system's adaptability allows for fine-tuning the model to serve a specific exam and incorporate relevant reference books as evaluation criteria. By uploading a textbook to the system, the evaluation criteria can be customized to align specific content and standards of the exam.

It enables customization, provides consistency and unbiased evaluation by finetuning a large language model—Llama2 [1]—and using a Retrieval Augmented Generation pipeline. Apart from this adaptability, which includes further finetuning of the model to bend it towards exams, this also includes the possibility of including certain reference books as evaluation criteria. The idea is that one could upload a textbook and get evaluation criteria that would be attuned to certain kinds of knowledge and standards that an exam should reflect.

This technique uses a fine-tuned large language model, Llama2, and a Retrieval-Augmented Generation pipeline. It gives high accuracy and reliability to the correction. The automated solution can not only allow more efficiency in the assessment process but also ensures fairness and frees educators to conduct more meaningful tasks that enhance the learning experience and skills.

The Overall End-to-End Machine Learning Workflow shows huge strides made toward innovative assessment procedures within the educational sector. In automating the evaluation of handwritten answer scripts, valuable time for educators is saved, biases are reduced, and students are given a more objective and reliable grading system. This innovation improves efficiency in the assessment procedure and takes us closer to realizing a much fairer and more effective educational environment.

## II. Motivation

### *A. Time Constraints and Efficiency*

Traditionally, the review process for handwritten answer scripts takes a long time, more than two or three weeks. This ultimately causes a delay in the announcement of the exam's final results, which affects students' academic schedules. With AI, the evaluation process will now be automated, significantly cutting down on the amount of time required for grading. By processing and evaluating the scripts using an automated system, it takes a fraction of the time compared to traditional human techniques. This speeds up the turnaround time for the results, assisting educational institutions in meeting deadlines.

### *B. Consistency and Fairness*

Grading answer scripts with large language models brings a new dimension to the process of grading in terms of uniformity and fairness. Unlike human graders, whose evaluations may be tilted by subjective biases or inconsistencies of various nature, LLMs have predefined criteria and algorithms to work on, making sure that each script is compared based on exactly the same standards. These models grade answers against a large amount of information, using established rubrics that greatly reduce variability in scoring. Fairness in this count means LLMs provide consistency in evaluation across all scripts, hence making the grading process more objective and reliable. This consistency goes a long way in safeguarding the integrity of academic assessments while ensuring each student is assessed on equal terms with others.

### C. Cost Effectiveness and Resource Optimization

Large Language Models (LLMs) involve cost-effectiveness and resource efficiency in evaluating answer scripts. Generally, evaluating exams involves huge resources and time wastage that results in a very expensive and inefficient process, particularly at its peak times during exams. In contrast, LLMs grade by automating the process, which reduces the need for large manpower and minimizes the time taken for large-scale grading. This not only saves operational costs but allows scalable evaluation; in this respect, a single model can handle thousands of scripts at a go. It therefore enables institutions to optimize their utilization of technological resources and channel their budgets more effectively into other important areas of education in order to ensure their overall operational efficiency and effectiveness.

## III. Existing Solutions

The answer scripts are evaluated by using various existing technologies and methodologies such as Optical Mark Recognition, Optical Character Recognition, and Automated Essay Scoring. Each of these systems has their own inherent flaws and shortcomings. Thus, they fail in terms of the versatility and comprehensive assessment abilities that manual correction offers.

### A. OMR

OMR stands for Optical Mark Recognition, a process used to capture data human-marked from documents such as surveys and tests. It is very effective in processing large volumes of forms quickly and precisely since it efficiently reads marks made with a pencil or pen on predefined positions on the paper. The major advantages of OMR are its speed, accuracy, and cost-effectiveness in standardized tests and questionnaires. However, some of the risks associated with this include perfect marking; any deviation, such as lightly marked marks, may result in errors. Another limitation of OMR is that it is limited only to multiple-choice formats, which cannot allow assessment of open-ended responses or solutions to complex problems, hence rendering its applicability quite limited in more diverse and nuanced assessment contexts.

### B. OCR

Specifically, with the enhancement by CNN algorithms, OCR can quite effectively be used in assessing objective-type questions, including MCQs, fill-in-the-blank responses, and matching items. What this does is combine the capacity of OCR in digitizing text with the capability of CNNs in undertaking pattern recognition and analysis, thus providing a boost to the accuracy and efficiency of automated assessment tools for these formats of questions. It may be less effective when the text includes much complex formatting, diagrams, or symbols to evaluate technical or scientific answers.

### C. Other Existing Technologies

Alikaniotis, D., Yannakoudakis, H., & Rei, M. (2016). The paper is concerned with an investigation into the use of neural networks for automatic text scoring, where in this case, it delves into deep learning methods and their advantages over traditional techniques in modeling complexities of language in student responses. [2]

Olowolayemo et al. 2018: The authors focus on short answer scoring for English grammar with the application of text similarity measurements; the methods improve the accuracy of automated grading systems.[3]

Selvi, P., & Banerjee, A.K. (2010): The Automatic Short-Answer Grading System ASAGS tries to meet the growing need for giving immediate feedback to students by comparing answers given by a student against reference models, which makes the process of grading faster and more efficient. [4]

Kadupitiya, J.C.S., Ranathunga, S., & Dias, G. (2016): Introduce measures of Sinhala short sentence similarity to evaluate short answers by corpus-based similarity techniques for improvement in the accuracy of assessment. [5]

Pribadi, F.S., Adji, T.B., Permanasari, A.E., Mulwinda, A., & Utomo, A.B. (2017): A study on automatic short answer scoring using word overlapping methods; endorses a basic method that proves practical and effective in assessing student answers. [6]

Chakraborty, U.K., Konar, D., Roy, S., & Choudhury, S. (2016): In this paper, an intelligent fuzzy spelling evaluator is proposed by the authors for any eLearning environment. The authors try to improve the assessment of spellings in an accurate and efficient manner in the journal Education and Information Technologies.[7]

Rababah and Al-Taani, 2017: The paper describes an automated scoring methodology for essays, which are short answers in Arabic. It has applied natural language processing techniques in this respect to improve reliability in grading. [8]

Vij, Tayal, and Jain, 2020: The authors use text similarity measures with WordNet graphs for automatic evaluation of short answers and have come up with results with high-precision grading. Wireless Personal Communications. [9]

Lajis, A., Nasir, H., & Aziz, N.A. (2019): The approach measures higher order thinking skills using short free text responses; thus, it encapsulates the requirement for holistic assessment tools within an educational context [10]

Menini, S., Tonelli, S., De Gasperis, G., & Vittorini, P. (2019): In this paper, a simple method is proposed to perform automatic short answer grading, which normally presents many problems for correctly evaluating the complex response of the student. [11]

Ratna et al. 2019: Benchmarks of Latent Semantic Analysis against Winnowing algorithms for the grading of Japanese short essay answers, aiming at the enhancement of effectiveness for an overall set of automated grading systems.[12]

Süzen et al. (2020) use text mining methods to grade and give feedback to short answers automatically, thus increasing the precision and reliability of grading systems.[13]

## IV. Methodology

Answer script evaluator works in collaboration with the examination unit and the evaluation center. It evaluates handwritten answers efficiently, consistently, and fairly. Here's an overview of how it works in detail:

### A. Exam Unit Side

Step 1: Prepare the set of questions to be evaluated.
Step 2: Upload the recommended textbook to the RAG pipeline and find the context of all the questions asked.
Step 3: Fine-tune the model Llama2 for the current examination.
Step 4: Deploy the model and integrate with the already existing UI.

### B. Evaluation Center Side

Step 1: Scan the answer scripts using a mobile camera or a scanner.
Step 2: Upload the answer script to the website and wait for the results.
Step 3: The handwritten answer script is converted to text using Google Cloud Vision API.
Step 4: The text is passed to our LLM, Llama2 with a prompt and the output is a score from 1-5.
Step 5: The RAG pipeline encompasses the context of the book related to each question and helps verify if the answer is appropriate and if the generated score is accurate.

## V. Technologies and Working Principles

Our model is a fine-tuned Large Language Model called Llama2, particularly used for the evaluation of handwritten exam scripts. It also uses a Retrieval-Augmented Generation pipeline to show relevant textbook pages. Deployed on AWS SageMaker for scalability, the solution integrates with a REST API via an AWS Lambda function for seamless operation.

### A. Dataset Curation

The curated dataset consists of 116 records, created by gathering questions from the Operating Systems course, along with student answers and corresponding grades (1 to 5) assigned by the teacher. The existence of this comprehensive dataset allows evaluation model and a RAG system to work together for accurate, rich in context assessment. This dataset is freely accessible on hugging face hub. Fig. 1 shows the dataset card on the hugging face website.

1. Question: Contains the question which is expected to be answered by the student.
2. Answer given by the student.
3. Grade given by the teacher for the corresponding answer.
4. Temp: The input format for the LLM model excluding Llama2.
5. Text: The input format for fine tuning the LLM model Llama2.
6. Context: This is created for the RAG pipeline.
7. Text: This is the updated input format for fine tuning the LLM model Llama2 after introducing the RAG pipeline.

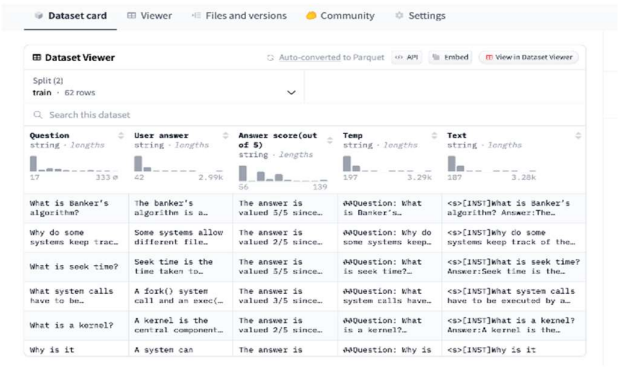


Fig. 1. Dataset used for fine tuning

### B. FineTuning

For this proposed task, the proposed solution makes use of the Llama2 model since it is open source and exhibits high accuracy. The Nous Research chat-based variant improves the use of prompt engineering to enhance the evaluation of answer scripts. The model is advanced in language understanding and adaptation, making it perfect for assessing student replies accurately. This provides a robust base for our evaluation system and allows assessments that are not only precise but also context-sensitive.

First, the following methods to adapt the Llama2 model with 7 billion parameters onto a T4 GPU must be run. Full fine-tuning on the T4 GPU is not possible due to the constraint of 16 GB of VRAM. The use of QLoRA, which are some parameter-efficient fine-tuning techniques that make it possible to fine-tune large models like Llama2 with reduced VRAM usage but without performance loss. To finetune this model the QLoRA which is the quantizied version of LoRA is used.

Quantization refers to the process where the weights of a model move from high to low precision, such as 32-bit floating-point to 4-bit. This allows for a much smaller model size, making it more viable on most GPUs. Particularly, QLoRA focuses on maintaining performance while the precision is being reduced. Quantization reduces memory usage during fine-tuning.

Traditional fine-tuning requires the adjustment of all parameters; this process is highly memory-consuming. Instead, techniques that are parameter-efficient, like QLoRA, are used. QLoRA involves either the tuning of only a small fraction of the parameters or the addition of a few new parameters that will be trained while keeping the rest of the model's parameters fixed. In this way, this method drastically reduces the number of parameters that need to be stored and updated in memory, hence optimizing VRAM usage.

After quantization and fine-tuning with optimized parameters, the model is trained on a comprehensive dataset of questions, student answers, and teacher grades. During training, some parameters are adjusted to help the model better align with the context and grading patterns in the dataset. A pipeline is created to automate the evaluation process. The pipeline leverages a fine-tuned text generation model, which assesses the user's answer based on the context and grading patterns in the dataset. The sample pipeline is illustrated below:

*prompt = '''Question: What is caching? Answer: Paging is a technique of memory management. \nEvaluate this answer*

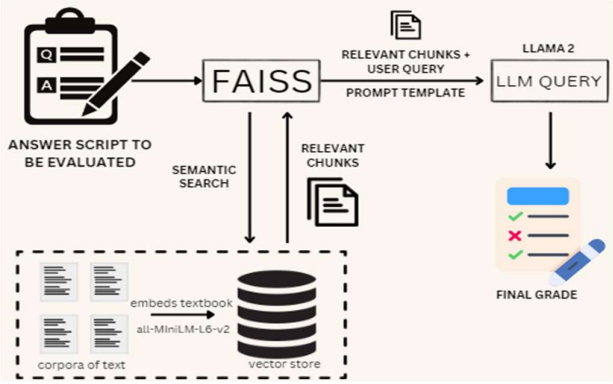

Fig. 2. Flowchart depicting the working of RAG.

*given by the user with reference to the question and give a score from 1-5 and be conservative in giving marks.'"*

```
pipe = pipeline(task="text-generation",
model=model,tokenizer=tokenizer,
max_length=300)
result = pipe(f"[INST] {prompt} [/INST]")
```

C. *RAG* (Retrieval Augmented Generation)

Retrieval Augmented Generation is an AI framework for enhancing LLM response quality where the model is grounded on external sources of knowledge to provide supplementary information for that represented internally in an LLM. RAG combines retrieval-based methods with generative models, ensuring that answers are grounded in the specific content of the embedded textbook. This leads to more precise and contextually accurate evaluations by our model. This study makes use of the all-MiniLM-L6-v2 model whose purpose is to generate high-quality embeddings efficiently. After which FAISS is used, with which an efficient semantic search can be performed to retrieve the most relevant content from a textbook based on the similarity of embeddings, thus enhancing the accuracy and speed of answer script evaluations. Given below is a figure which represents the workflow. Fig. 2 gives a flowchart for the RAG pipeline.

*D.* Deployment

Llama2 is a billion-parameter model, so running it efficiently would require huge computational resources. Partly, huge RAM and probably GPUs are needed. Most computers locally would, therefore, not have the power to run this model accordingly. AWS SageMaker is used to deploy the model. SageMaker provides the size and required infrastructure to manage the computational load.

Scalability: SageMaker allows resources to be scaled up or down on demand. This means it handles varying loads—from low to high—with the system still performing well on tests.

Accessibility: Global deployment ensures that the system can provide reliable grading regardless of geographical location and that it can be accessed from any institutional server in the world.

The following steps can be followed for deployment of the fine-tuned version of Llama2 on the AWS SageMaker.

1. In the first step, a model was loaded from Hugging Face, and a SageMaker Notebook instance was created for its deployment. The right instance type for executing the correct computations by the model was set up for the Notebook instance. This step is key to optimal performance and scaling at deployment.
2. An endpoint is created for the instance to be accessible through a Lambda function.

```
predictor = huggingface_model.deploy(
initial_instance_count=1,
instance_type="ml.m5.large")
```

3. Lambda Function to handle requests REST API: A Lambda function is created that makes requests against the REST API and handles all processing using the SageMaker endpoint. This function will run as an interface to obtain data from the API, call the SageMaker endpoint, and return the result to the API. The Lambda function was tested on all possible scenarios regarding functionality and reliability by verifying that it could correctly and efficiently handle requests. This testing phase is critical in proving that the function does what it is supposed to do before one can get into other phases of a study.
4. In this step, a REST API was configured to trigger the Lambda function. A POST request is used to invoke the function. The invoke URL for this trigger is copied to facilitate subsequent integration and testing. The test option from API Gateway will be used to test if the API trigger works fine. This testing phase ensured that the API correctly calls the Lambda function and handles requests as expected, thus validating the integration between API Gateway and the Lambda function before going further on deployment. The complete architecture is mentioned in Fig. 3.

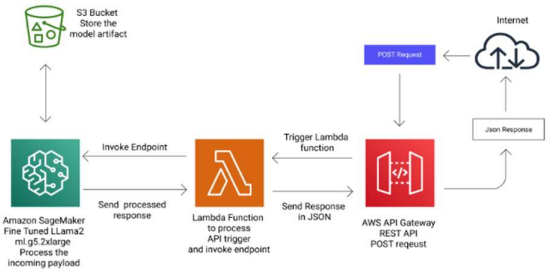

Fig. 3. AWS architecture.

## VI. Results

In evaluating the performance of AutoGrader, the scores predicted by the model are compared against the ground truth values, which are the scores provided by teachers for the semester-end examinations. The comparison was performed using two key metrics: Mean Absolute Error (MAE) and Mean Squared Error (MSE).

The Mean Absolute Error, which measures the average magnitude of the prediction errors, was calculated as 0.50 units. This value indicates that, on average, the scores predicted by our model deviated from the teacher-assigned scores by 0.50 units. The results demonstrate that our AutoGrader performs with a high level of accuracy. The MAE of 0.50 suggests that, on average, the model's grading is reasonably close to the

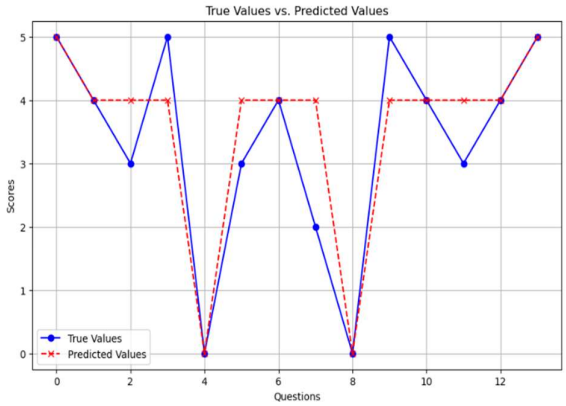

Fig. 4. Plot between the ground truth values and the predicted values.

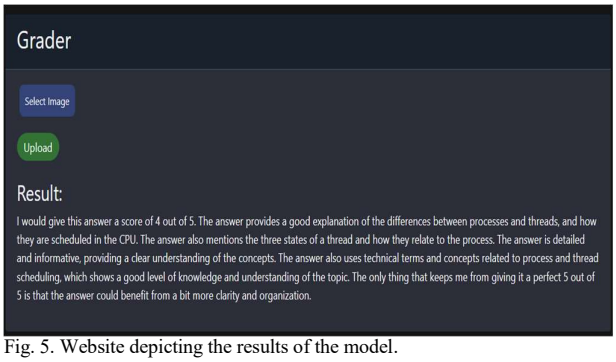

Fig. 5. Website depicting the results of the model.

scores assigned by teachers. The Mean Squared Error, which emphasizes larger errors by squaring the differences before averaging, was determined to be 0.61 units. The MSE value indicates that there are relatively few large discrepancies affecting the overall performance as compared to the MAE.

Additionally, the AutoGrader is deployed as a live website hosted on Amazon SageMaker and AWS Lambda. This deployment allows users to interact with the system and see the grading in action, the demo of the website is given below in Fig. 5.

## VII. Conclusion and Future Scope

The education sector will undergo a transformation thanks to our AI-based automatic answer script evaluator, which will assist with answer script evaluation at all educational levels. The approach offers a scalable and effective means of evaluating theory-based test answer scripts, ranging from high school board exams to undergraduate and graduate exams.

The technique used here is especially suitable for a Software as a Service (SaaS) offering that handles large amounts of response scripts. Every document that is examined will be charged a price in our proposed business model, providing a distinct and scalable source of income. Any educational setting, examination boards, and other organizations needing dependable and effective test grading might use this paradigm.

Currently the model is incapable of recognizing figures in the answer script. Also, since LLM's are bad at reasoning, this model cannot evaluate questions based on logic and reason.

Future efforts will focus on evaluating diagrams, reducing hallucinations, and improving consistency. One of the major issues in LLM's is that there is a different answer for a same prompt. However, the model is still capable of generating output more consistent than human grading.

## References

[1] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. Canton Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, and S. Edunov, "Llama2 2: Open Foundation and Fine-Tuned Chat Models," arXiv preprint arXiv:2307.09288, 2023. [Online]. Available: https://arxiv.org/abs/2307.09288.

[2] Alikaniotis, D., Yannakoudakis, H. and Rei, M., 2016. Automatic text scoring using neural networks. arXiv preprint arXiv:1606.04289.

[3] Olowolayemo, A., Nawi, S.D. and Mantoro, T., 2018, September. Short Answer Scoring in English Grammar Using Text Similarity Measurement. In 2018 In- ternational Conference on Computing, Engineering, and Design (ICCED) (pp. 131-136). IEEE

[4] Selvi, P. and Banerjee, A.K., 2010. The auto- matic short-answer grading system (ASAGS). arXiv preprint arXiv:1011.1742.

[5] Kadupitiya, J.C.S., Ranathunga, S. and Dias, G., 2016.Sinhala Short Sentence Similarity Measures for Short Answer Evaluation based on Corpus-Based Similarity.

[6] Pribadi, F.S., Adji, T.B., Permanasari, A.E., Mul- winda, A. a Utomo, A.B., 2017, March. Auto- matic short answer scoring using word overlapping methods. In AIP Conference Proceedings (Vol. 1818, No. 1, p. 020042). AIP Publishing LLC

[7] Chakraborty, U.K., Konar, D., Roy, S. and Choud- hury, S., 2016. Intelligent fuzzy spelling evaluator for eLearning systems. Education and Information Technologies, 21(1), pp.171-184.

[8] Rababah, H. and Al-Taani, A.T., 2017, May. An automated scoring approach for Arabic short answers essays questions. In 2017 8th International Confer- ence on Information Technology (ICIT) (pp. 697- 702). IEEE

[9] Vij, S., Tayal, D. and Jain, A., 2020. A machine learning approach for automated evaluation of short answers using text similarity based on WordNet graphs.Wireless Personal Communications, 111(2), pp.1271- 1282.

[10] Lajis, A., Nasir, H. and Aziz, N.A., 2019, August. NCI Evaluation: Assessment of Higher Order Thinking Skills via Short Free Text Answer. In 2019 IEEE International Conference on Smart Instrumentation, Measurement and Application (ICSIMA) (pp. 1-5). IEEE.

[11] Menini, S., Tonelli, S., De Gasperis, G. and Vittorini, P., 2019, November. Automated Short Answer Grading: A Simple Solution for a Difficult Task. In CLiC- it.

[12] Ratna, A.A.P., Santiar, L., Ibrahim, I., Purnamasari, P.D., Luhurkinanti, D.L. and Larasati, A., 2019, October. Latent Semantic Analysis and Winnowing Al- gorithm Based Automatic Japanese Short Essay Answer Grading System Comparative Performance. In 2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST) (pp. 1-7). IEEE.

[13] Süzen, N., Gorban, A.N., Levesley, J. and Mirkes, E.M., 2020. Automatic short answer grading and feedback using text mining methods. Procedia Computer Science, 169, pp.726-743.