ICIC Express Letters
Part B: Applications
Volume **14**, Number **10**, October **2023**

ICIC International ©2023 ISSN 2185-2766
pp. **1001–1009**

WEB-BASED APPLICATION FOR AUTOMATED ESSAY SCORING USING DEEP LEARNING TO SUPPORT ASSESSMENT PROCESS IN E-LEARNING

Kadarisman Kadarisman[1], Retno Kusumaningrum[2,*]
Linggar Maretva Cendani[2], Selvi Fitria Khoerunnisa[2]
Sukmawati Nur Endah[2], Khadijah Khadijah[2], Rismiyati Rismiyati[2]
Priyo Sidik Sasongko[2], Afriani Afriani[3] and Irdha Yusra[4]

[1]Faculty of Education and Teacher Training
[3]Faculty of Law, Social and Political Science
Universitas Terbuka
Jalan Cabe Raya, Pondok Cabe, Pamulang, Tangerang Selatan 15437, Banten, Indonesia
{ kadarisman; afriani }@ecampus.ut.ac.id

[2]Department of Informatics
Faculty of Science and Mathematics
Universitas Diponegoro
Jl. Prof. Soedarto, SH Tembalang, Semarang 50275, Central Java, Indonesia
{ linggarmc; selvifitria }@students.undip.ac.id; { sukmane; priyosidiksasongko }@lecturer.undip.ac.id
{ khadijah; rismiyati }@live.undip.ac.id
*Corresponding author: retno@live.undip.ac.id

[4]Department of Management
Universitas Negeri Padang
Jalan Prof. Dr. Hamka. Rd, Air Tawar, Padang 25131, West Sumatera, Indonesia
irdhayusra@fe.unp.ac.id

Received November 2022; accepted February 2023

Abstract. *The Automated Essay Scoring (AES) application was designed to support the learning process and assessment of e-learning platforms. An essay question is a question that can capture students' analytical and synthesis abilities. However, the assessment of this form of questions is still done manually, and it is not the flexible learning concept of e-learning. Therefore, this study aims to develop an application that automatically scores essay questions. The application is a web-based application with system development using an iterative waterfall model – the automatic scoring employed several deep learning models, i.e., IndoBERT as the pre-trained word embedding model and CNN-LSTM with Mean Over Time as the model for automated final scoring. Three implemented testing mechanisms are unit testing using black box testing, system integration testing and model performance testing based on QWK and loss value. All test cases that have been tested are indicated to be successful. In addition, the deep learning model performance results in a loss value of 0.04 and QWK of 0.451.*
Keywords: Automated essay scoring, Assessment, CNN-LSTM with mean over time, E-learning, IndoBERT, Web-based application

1. **Introduction.** The COVID-19 pandemic has emerged since 2019 and continues to spread to this day which has resulted in changes in various fields. One of them is the world of education, significantly higher education. Initially carried out in the classroom, the teaching and learning process is now moving to a virtual room by utilizing various technologies, such as video conferencing applications and e-learning platforms [1]. In addition, the concept of open education at the higher education level also continues to grow.

DOI: 10.24507/icicelb.14.10.1001

Various groups increasingly demand it, so the reliability of the e-learning platform in the learning and assessment process is a vital aspect to pay attention to.

E-learning is not a new technology but a technology that has been developed for several decades. However, one of the things that still leaves homework is the assessment process, where assessment plays an essential role in the educational process [2]. The scoring system generally used is in the form of multiple-choice questions because of the ease of implementing and assessing the results of tests or managing student responses [3]. However, the grading system has several areas for improvement, i.e., the results of a high score do not necessarily reflect students' actual abilities because it is possible for students to provide speculation or guesswork answers. In addition, the form of this question tends to be unable to capture students' analytical and synthesis skills. Hence, the assessment system using essay questions is important to be developed further.

The assessment system for essay questions provided on the e-learning platform is still limited to the teacher's manual assessment process, which results in students not being able to get assessment results and feedback in real time. It is not following the concept of e-learning, where students are expected to get a flexible learning experience that can be accessed anywhere, at any time, by anyone.

Therefore, it is necessary to develop an Automated Essay Scoring (AES) application for Indonesian-language documents. Various studies have been developed, but they still generally use the Bag-of-Words (BoW) concept approach, where a document is considered a bag of words or ignores the semantic context of a sentence. These studies, among others, utilize the N-grams feature and string similarity measure [4] or the N-grams feature and the latent semantic analysis algorithm [5,6].

There are two drawbacks of BoW, including the n-grams model, i.e., (i) it has the curse of dimensionality issue since the vocabulary size specifies the vector size; (ii) it does not consider the semantic relation between words. Therefore, this study applies IndoBERT as a word embedding technique because it can overcome the weaknesses of BoW. Furthermore, this study implements the CNN-LSTM method with a mean over time for measuring the similarity between the answers and determining the score. The CNN-LSTM is expected to be able to perform high-level feature extraction and the process of relationship learning between essay answers and scores automatically in a series of processes. In addition, LSTM is one of the recurrent models that can cope with the problem of vanishing gradients more effectively. Compared to feed-forward neural networks, recurrent models are more robust and capable of learning more complex patterns from data.

The remainder of this paper is organized as follows. Section 2 describes the web-based application development, i.e., the employed iterative waterfall model for the software development life cycle. User interfaces for three main processes, including a form for input data, a form of initial scoring, and a result form related to final scoring, are discussed in Section 3, followed by the evaluation process. The conclusions and future work are outlined in Section 4.

2. **Application Development.**

Application development is the developing process of an application or software for a particular purpose. The software development process aims to ensure that the end product can meet the software development purpose and fit into the overall budget. Systems Development Life Cycle (SDLC) is a common term summarizing software or application development stages [7,8].

Furthermore, the simple and idealistic model of SDLC is the classical waterfall model. This model divides application development stages into six stages: requirements elicitation, design, implementation, testing, deployment, and maintenance [9]. However, the classic waterfall model is challenging because there is no feedback path, so there should be no error at any stage of the work. Therefore, this study employed an iterative waterfall model involving a feedback path.

The iterative model conducts the development process in a cyclic manner, i.e., (i) the initial development is a small-scale development in which all steps are followed sequentially; (ii) afterwards, more features and modules are designed, tested, and appended to the project in successive iterations [10]. The illustration of the iterative waterfall model can be seen in Figure 1.

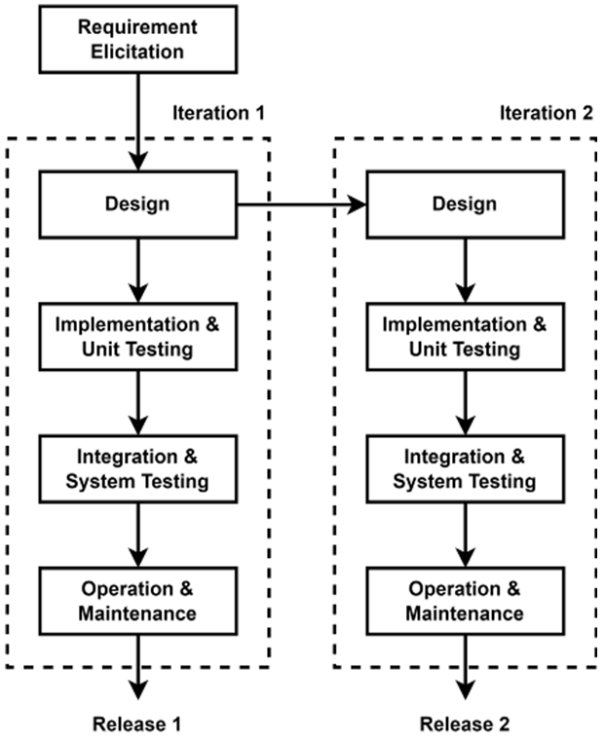


FIGURE 1. The employed iterative waterfall model

The first stage is the requirement elicitation process. This stage is vital in the software development process because failure in carrying out the requirements elicitation process might lead to acquiring low-quality user requirements and failure of software development projects [11]. We employ a use case diagram as the artefact for software requirement elicitation. Use cases describe user and system interactions to find user needs. Use cases are commonly used for specifying functional requirements [12]. Figure 2 shows the use case diagram for the developed AES application.

Figure 2 explains that the AES system has two users: administrators and lecturers. The administrator has the main task of "managing question statistics". The questions referred to here are not only questions but also include a set of student answers that will be given an automatic assessment. To be able to do this, the admin must first "log in" to the system. Furthermore, the lecturer has the main task of "managing questions", whereas previously, he had to "log in" to the system first. The cases of "opening auto-grading results", "downloading automated scoring results", and "final scoring" do not make sense without "managing questions" that have been run. "Final scoring" here is the process of evaluating a set of answers to a respected question that can be accomplished if the "log in" process and "initial score" have been performed.

In addition, we also provided the activity diagram. An activity diagram describes how a system is used in terms of activities [13]. In web application development, each activity in the activity diagram can be implemented as one web page or a group of web pages in a subsite or a functional cluster. Since opening automated scoring results, downloading automated scoring results, managing question statistics, and input data involve a simple activity, we do not display activity diagrams for these four use cases. The main activity diagram for the developed AES application can be seen in Figure 3.

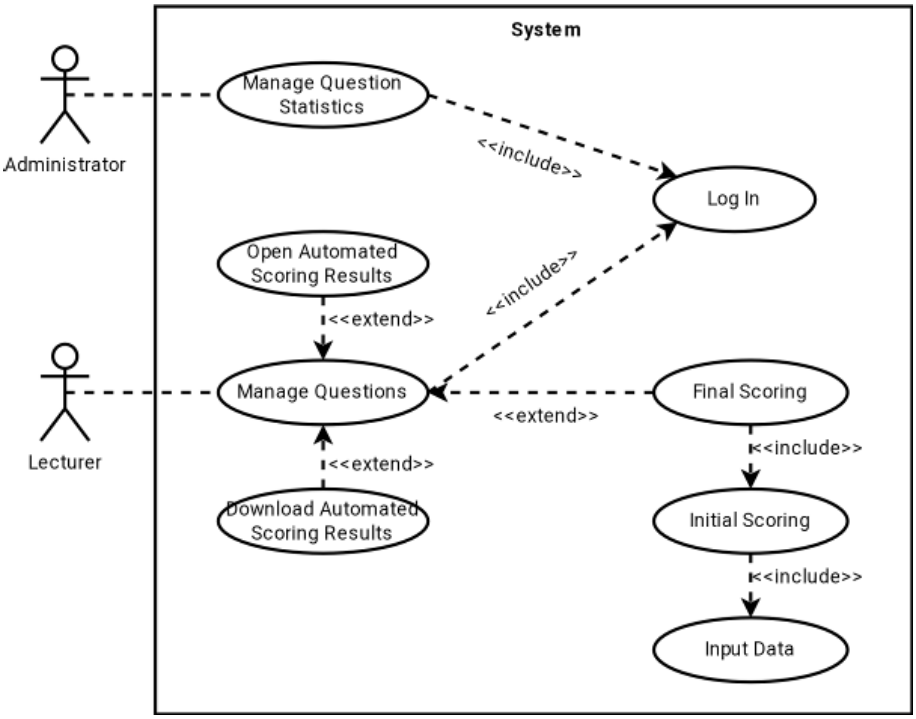


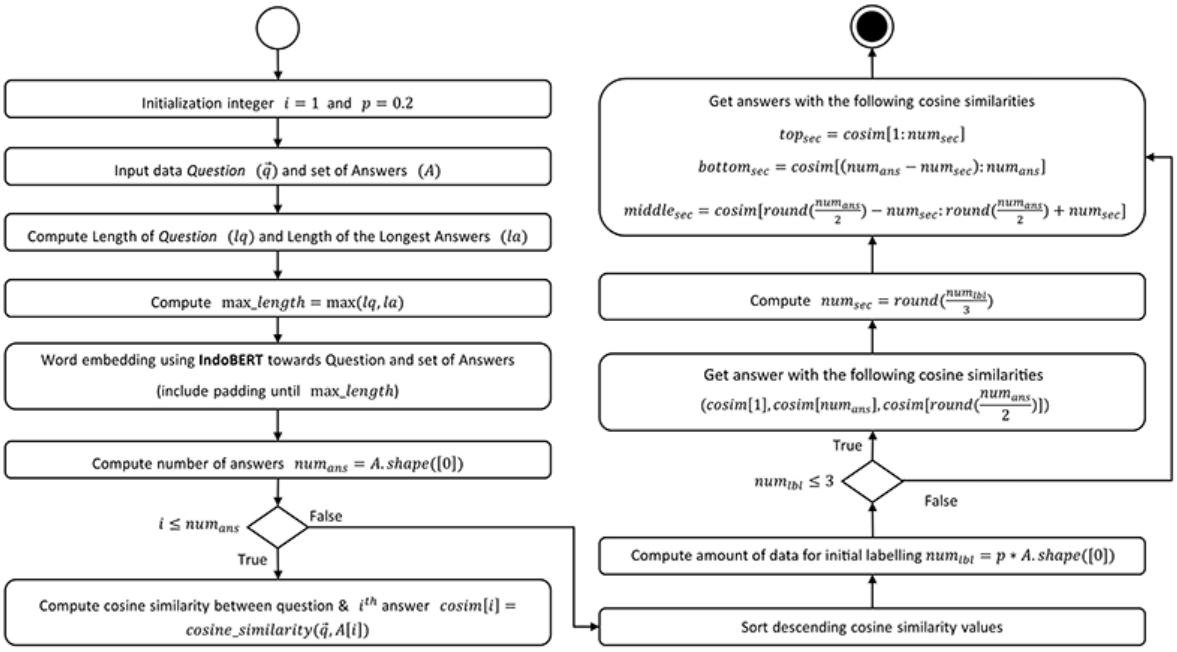
FIGURE 2. Use case diagram of the developed AES application

As depicted in Figure 3, it can be seen that there are two deep learning models employed in this study. The first model is IndoBERT [14], the pre-trained word embedding model. IndoBERT was chosen because it has rich features but is a simpler model. The second model is CNN-LSTM with Mean Over Time [15] as the model for automated final scoring.

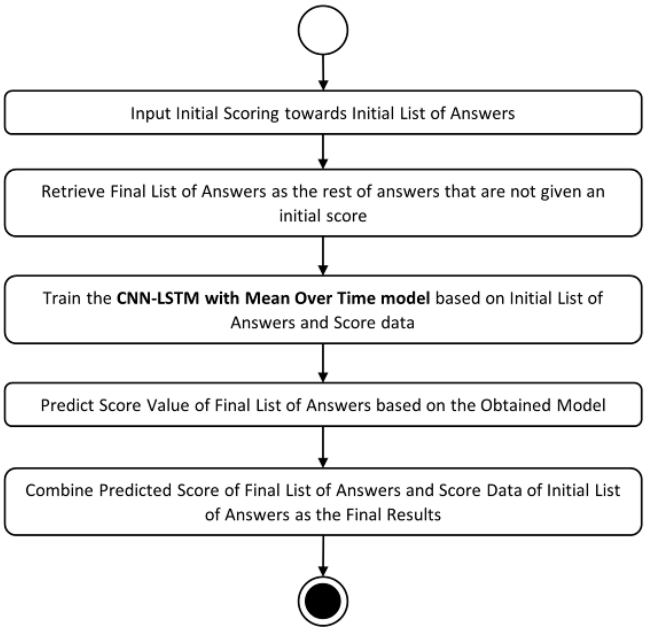
Based on Figure 3, the system input is a list of questions and answers. Input will be embedded using IndoBERT. Then the answers will be sorted based on similarity to the question, and 20% will be taken as a list for initial scoring. The 20% decision process is implemented by taking 6% of the answers that are most similar to the question, 6% of the answers that are the least similar to the question, and 8% of answers that are similar in the middle ranking. Furthermore, a set of selected answers and the initial score is used as training data to form an automatic scoring model. The model was built using CNN-LSTM with Mean Over Time.

The design process consists of three things: designing the user interface, database, and algorithmic functions. There are three main pages: the AES application's landing page, admin dashboard, and lecturer dashboard. The detailed process of automated essay scoring is accommodated by using three modal windows, including input data, initial scoring for selected answers, and the result of the automated essay scoring process towards the rest of the data. In addition, we also design several modal windows for several notifications. Furthermore, the database design stage produces three tables: the questions table, the answer table, and the user table.

This study has employed deep learning models implemented using the Python programming language. In building user interfaces, HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), Javascript and PHP (Hypertext Preprocessor) were used as the basis for making web-based applications. While the database in the Transformers 1.0 system is implemented using SQLite. To connect between models and web-based interfaces, the development of the AES application requires a Flask library. Flask is a web development framework from the python library that will make the developed web more structured and can manage behavior easily.



(a) Initial scoring



(b) Final scoring

FIGURE 3. Main activity diagrams of the developed AES application

After the implementation stage, the next stage is testing. This test assesses whether the overall system can meet the design requirements. The tests will be based on unit testing by using black box testing. Subsequently, we performed system integration testing, which means testing will focus on automation and dependability between existing components. In addition, this study also evaluates the performance of the implemented model using QWK and loss value. The QWK value calculates by comparing the value generated by the developed application with the value given by the human annotator. The second iteration aims to complete the process of requesting accounts and resetting passwords for lecturers, so it is separate from the automated essay-scoring process.

3. **Result and Discussion.** Two actors are involved in the application being developed, i.e., lecturer and admin. The following are various features for lecturers, including data input that will be automatically scored, initial score input for selected sample data, and a display to see the automated final scoring results of all the submitted data. These features can be seen sequentially in Figures 4(a), 4(b), and 4(c).

As previously explained, the automatic essay scoring process starts with the lecturer submitting data, including 1 question and a collection of answers stored in a spreadsheet format. Figure 4(a) shows the application interface for data input. Furthermore, the application will select as much as 20% of the uploaded answers to be displayed to the users. The collection of selected answers is in the future referred to as the initial list of answers. The interface for displaying the initial list of answers can be seen in Figure 4(b). The picture shows that each answer is equipped with a text box to provide an initial assessment showing each answer's accuracy level. Subsequently, Figure 4(c) shows the system's final score, which is automatically determined for each answer.

As mentioned before, we performed three testing mechanisms: unit testing using black box testing, system integration testing and model performance testing based on QWK and loss value. Mean square error function is used for calculating loss of model as shown in Equation (1), where n is number of data, Y_i is actual value, and $\hat{Y}_i$ is predicted value. Then, QWK is calculated starting with creating $N \times N$ histogram matrix O (O_{ij}), where i is the human rating and j is system rating. After that, N by N matrix of weight (W_{ij}) is calculated by Equation (2). Matrix E is calculated with assuming that value not related. Matrices O and E are normalized and QWK is calculated as following Equation (3).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \tag{1}$$

$$W_{ij} = \frac{(i-j)^2}{(N-1)^2} \tag{2}$$

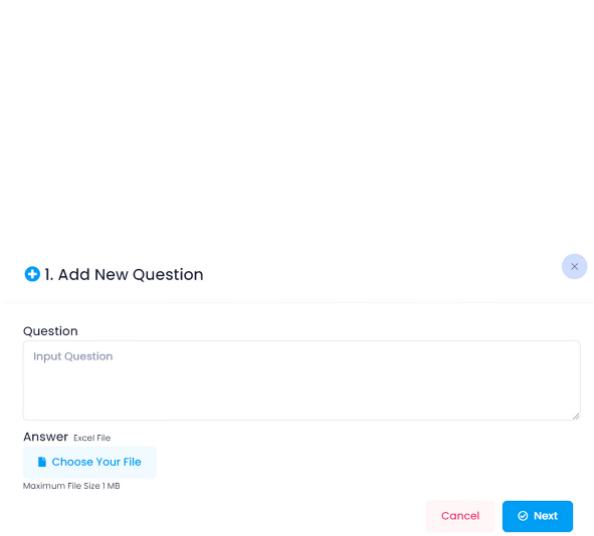
$$\kappa = 1 - \frac{\sum_{ij} W_{ij} O_{ij}}{\sum_{ij} W_{ij} E_{ij}} \tag{3}$$

Since the developed application was web-based, we employed several common browsers, including Google Chrome, Microsoft Edge, Mozilla Firefox, and Opera Browser. Whereas the hardware specification for the testing process is as follows:

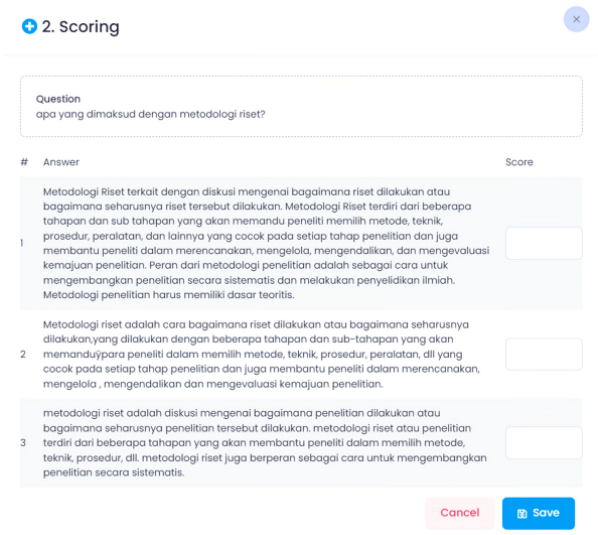
- Processor: Intel®Core™ i7-6500 CPU@2.5GHz-2.59GHz
- Memory: 12 GB
- Storage: 1 TB Full SSD
- Graphics: NVIDIA®GeForce®940M 2GB

The testing results stated that the application process was running according to the design, and the existing button functions were under the need. All test cases that have been tested are indicated to be successful. It indicates that this application has met the software requirements that have been set. In addition, the tests concluded that the application could give scores to the set of answers that were input based on related questions. We test the performance of the applied deep learning model using two datasets: dataset 1 is the dataset question A from the Ukara dataset [16], and dataset 2 is the dataset created by researchers related to the research methodology topic. Table 1 shows the obtained QWK and loss value.

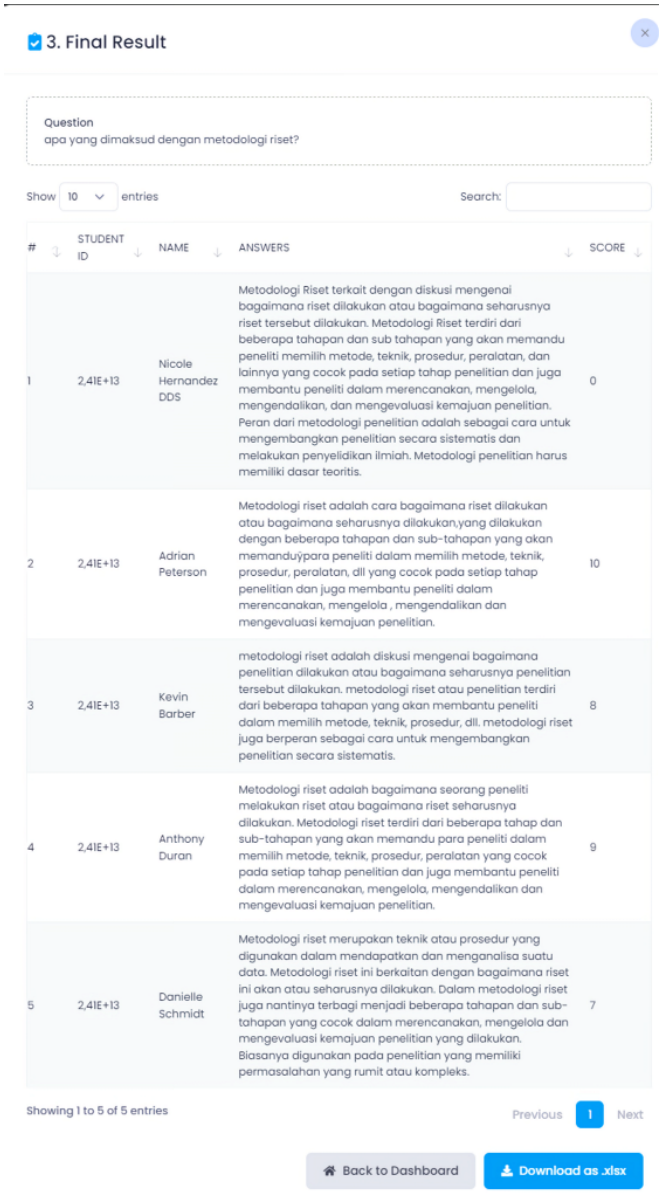
4. **Conclusions.** The proposed automated essay scoring application has been successfully implemented and used in an actual process. Essay questions can be scored automatically using this application, and the user only needs to score several answers chosen by the application as an initialization list. The number of initial values selected in this study



(a) Input data



(b) Initial scoring for selected answers



(c) Final scoring result

FIGURE 4. User interface of automated essay scoring activity

TABLE 1. The QWK and loss value

Dataset	Loss value	QWK
Dataset 1	0.035	0.438
Dataset 2	0.045	0.464
Average	0.04	0.451

is 20% of the input answers. Subsequently, the remaining answers will be scored automatically by the system. The test results obtained a QWK value of 0.451 with a loss value of 0.04, indicating that the system successfully gave value to the essay. Because the final result in the form of a QWK score is still low, around a value of 0.451, improvements can be made for further research by developing a model based on other more advanced deep learning models, such as applying the attention mechanism since the attention mechanism can assign different importance to the different elements of the input sequence and gives more attention to the more relevant inputs.

Acknowledgment. This work was supported in part by the Universitas Terbuka, Indonesia under the Research Grant Number B/650/UN31.LPPM/PT.01.03/2022. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] C. Coman, L. G. Ţîru, L. Meseşan-Schmitz, C. Stanciu and M. C. Bularca, Online teaching and learning in higher education during the coronavirus pandemic: Students' perspective, *Sustainability*, vol.12, no.24, 10367, 2020.

[2] M. J. Hazar, Z. H. Toman and S. H. Toman, Automated scoring for essay questions in e-learning, *Journal of Physics: Conference Series*, vol.1294, pp.1-14, 042014, 2019.

[3] Y. Zaindanu and M. T. Ihsan, The role of Google form as an assessment tool, *ELT: Critical Review of the Literature*, vol.1, no.1, pp.58-66, 2021.

[4] M. A. Fauzi, D. C. Utomo, E. S. Pramukantoro and B. D. Setiawan, Automatic essay scoring system using n-gram and cosine similarity for gamification based e-learning, *Proc. of the International Conference on Advances in Image Processing*, Bangkok, Thailand, pp.151-155, 2017.

[5] R. S. Citawan, V. C. Mawardi and B. Mulyawan, Automatic essay scoring in e-learning system using LSA method with n-gram feature for Bahasa Indonesia, *Proceedings of the 3rd International Conference on Electrical Systems, Technology and Information (ICESTI2017) – MATEC Web of Conferences*, vol.164, no.3, pp.1-17, 01037, 2018.

[6] E. S. Pramukantoro and M. A. Fauzi, Comparative analysis of string similarity and corpus-based similarity for automatic essay scoring system on e-learning gamification, *Proc. of 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Malang, Indonesia, pp.149-154, 2016.

[7] S. Ergasheva and A. Kruglov, Software development life cycle early phases and quality metrics: A systematic literature review, *Journal of Physics: Conference Series*, vol.1694, no.1, 2020.

[8] S. Gupta, J. Banga, S. Dabas and M. K. Bhatia, A comprehensive study of software development life cycle models, *Int. J. Res. Appl. Sci. Eng. Technol.*, vol.10, no.12, pp.354-358, 2022.

[9] G. Gurung, R. Shah and D. P. Jaiswal, Software development life cycle models – A comparative study, *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp.30-37, 2020.

[10] M. S. Gharajeh, Waterative model: An integration of the waterfall and iterative software development paradigms, *Database Systems Journal*, vol.10, pp.75-81, 2019.

[11] W. Behutiye et al., Management of quality requirements in agile and rapid software development: A systematic mapping study, *Information and Software Technology*, vol.123, 2020.

[12] M. Lounakoski, J. Puhto, H. Jalo, O. Torro and H. Pirkkalainen, Social extended reality-use case entities on property life cycle, *Journal of Information Technology in Construction*, vol.27, pp.512-528, 2022.

[13] F. Wang, UML diagram classification model based on convolution neural network, *Optik (Stuttg)*, 2022.

[14] F. Koto, A. Rahimi, J. H. Lau and T. Baldwin, IndoLEM and IndoBERT: A benchmark dataset and pre-trained language model for Indonesian, *Proc. of the 28th International Conference on Computational Linguistics*, Barcelona, Spain, pp.1-14, 2020.

[15] K. Taghipour and T. H. Ng, A neural approach to automated essay scoring, *Proc. of the EMNLP 2016 – Conference on Empirical Methods in Natural Language Processing*, Austin, TX, USA, pp.1882-1891, 2016.

[16] G. B. Herwanto, Y. Sari, B. N. Prastowo, M. Riasetiawan, I. A. Bustoni and I. Hidayatulloh, UKARA: A fast and simple automatic short answer scoring system for Bahasa Indonesia, *Proceedings of International Conference on Educational Assessment and Policy (ICEAP)*, Jakarta, Indonesia, vol.2, pp.48-53, 2018.