Transformers Based Automated Short Answer Grading with Contrastive Learning for Indonesian Language

1st Aldo Arya Saka Mukti
Dept. of Electrical and Information Engineering
Universitas Gadjah Mada
Sleman, Indonesia
aldoarya00@mail.ugm.ac.id

2nd Syukron Abu Ishaq Alfarozi*
Dept. of Electrical and Information Engineering
Universitas Gadjah Mada
Sleman, Indonesia
syukron.abu@ugm.ac.id

3rd Sri Suning Kusumawardani
Dept. of Electrical and Information Engineering
Universitas Gadjah Mada
Sleman, Indonesia
suning@ugm.ac.id

2023 15th International Conference on Information Technology and Electrical Engineering (ICITEE) | 979-8-3503-0446-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICITEE59582.2023.10317785

***Abstract*—The rapid development of technology has impacted various sectors, including education. These developments have enabled e-Learning to thrive, especially during the Covid-19 pandemic. Evaluating student performance and understanding in e-Learning is typically done through quizzes. However, these evaluations, especially in essay grading, still require manual effort. This can lead to exhaustion and introduce bias and inconsistency into the scoring process. To address this issue, one possible solution is to develop an automated short-answer grading system. This research explores large language model that has a general understanding of language. This model is then subjected to a finetuning process. Specifically, this study employs BERT model, with contrastive learning method to develop an automated short-answer scoring system and compare its performance with similar systems. The model is composed of two components, namely the model body which utilizes BERT variation and the model head which employs logistic regression. The model body is structured in a siamese architecture. The results demonstrate an improvement in model performance of BERT model with constrastive learning. When compared to the pretrained BERT and BERT with cosine similarity finetuning, the reduction in prediction MAE is 21.72% and 9.90%, while for the RMSE metric, it is 17.79% and 13.80%. The transformers-based model with contrastive learning achieves metrics of 0.191 for MAE and 0.231 for RMSE. These findings indicate the potential of using the contrastive learning method in transformers models to develop an automated short-answer scoring system.**

***Index Terms*—contrastive learning, transformers, automated short answer grading, e-Learning, large language model**

I. Introduction

Information technology advancements, notably e-Learning, delivered via the Internet, have significantly impacted education across various sectors [1]. The widespread adoption of e-Learning in both formal and informal educational settings reflects its benefits against traditional learning method, including flexibility, student-centered learning, collaboration, cost-effectiveness, and tailored learning options [2], [3].

The importance of adopting e-Learning in educational institutions has grown due to the Covid-19 pandemic. E-Learning helps schools continue teaching when face-to-face classes aren't possible [4]. This approach benefits teachers and students by making learning materials accessible, information dissemination effective, and assignments manageable [5] [6].

In the evolving education landscape, the fusion of deep learning and Learning Analytics (LA) has ignited a transformation. Deep learning's data analysis and pattern recognition, combined with LA's contextual insights, hold promise for enhancing education, including Intelligent Tutoring Systems (ITS) to automated grading [7]. A notable example is automated grading, where deep learning algorithms efficiently assess student work.

In e-Learning, assessing student progress usually involves assignments and quizzes, including essays. Manual essay grading can exhaust teachers, leading to biased evaluations and inconsistent results [8]. This fatigue also affects students, diminishing their perception of teacher support and impacting academic performance [9], [10].

One way to address the manual grading issue, particularly for short answer essays, is by using an Automated Short Answer Grading (ASAG) system. Several ASAG systems have been developed, such as Intelligent Essay Assessor (IEA), Project Essay Grade (PEG), and E-Rater. However, these systems have their respective weaknesses and are considered to no longer meet future needs [11].

Advancements in natural language processing (NLP) research, particularly in text similarity, provide opportunities for leveraging NLP techniques as a solution for ASAG

*Corresponding author. Email: syukron.abu@ugm.ac.id.

development. Moreover, models developed to tackle NLP problems are rapidly evolving. Transformers are state-of-the-art neural network models specifically designed to address natural language processing challenges [12]. Transformers can be used to obtain semantic similarity between two documents, considering not only the sentences or words composing the text but also the context of each document.

This research explores the application of contrastive learning methods and the use of transformer models to develop ASAG in e-Learning. The grading will be based on the extracted feature from the combination of embedding that represents the teacher's answer key and the student's answer. This research aims to examine the influence of contrastive learning methods on the performance of transformer models in predicting students' answer scores, which serve as the basis for automated grading, in comparison to previous methods.

II. Related Works

Several studies on automated essay grading systems have explored various methods, both traditional and modern ones, including machine learning. Previous research has explored the usage of text distance-based ASAG for the Indonesian language [13] [14]. This research mainly explored the use of TF-IDF to create vector representations with different weights based on how often a word shows up on a document. The score is then determined by the cosine similarity or Jaccard similarity between the teacher's key answer and the student's answer. The result of these text distance-based ASAGs has not been able to compete with human scoring. It also showed the need to also explore answer semantics, instead of relying only on lexical.

Further development of ASAG used text representation, such as the use of latent semantic analysis (LSA) [15] [16] and transformers models [17] [18] [19]. Ratna et al. [15] combined LSA and SVM, where SVM was used to filter and classify the topics of students' answers and then compared to the teacher's key answers. This research employs LSA to create a document matrix and TF-IDF to construct the vector representation. This document matrix is then used to calculate similarity. Citawan et al. [16] employed similar usage of LSA to calculate similarity but combined it with the n-gram feature.

Transformers-based ASAG used BERT models to create an embedding representation of both teacher's key answers and the student's answers. These embeddings are then fed into a regression layer to predict the student's answers. BERT-based ASAG used sentence embedding that was generated by feeding token embedding into a pooling layer to achieve better performance [17] [19]. Haidar and Purwarianti [17] experimented with various BERT variations and features used by the regression layer. This research showed the best result by using 'bert-base-multilingual-cased'. The features used to obtain the best performance in this research were the combination of teacher's key answers embedding, student's answers embedding, the absolute difference between embedding, and element-wise multiplication between embedding. Salim et al. [18] compared the performance between the ridge regression model and the BERT model for the ASAG task and showed that the BERT model has superior performance compared to the ridge regression model.

Various research also showed better performance by using a transformers-based model to compare text or document similarity. Peinelt, Nguyen, and Liakata [20] used BERT that combined topic modeling to achieve better performance in binary classification based on text similarity. This research showed that text topics serve as additional information for the model, especially in domain-specific cases. Mutinda et al. [21] experimented with BERT to compare Japanese medical documents and showed a high Pearson correlation score.

In conclusion, various approaches have been used to develop ASAG for the Indonesian language, both using text-distance and text-representation methods. These researches also showed an improvement going from text distance to the text-representation method. Other research also showed the usage of a transformers-based model to calculate the similarity between documents that are able to achieve superior performance compared to the previous method. This opened the possibility of choosing a transformers-based method for ASAG and improving it further. This research aims to improve the current transformers-based ASAG by using the contrastive learning method.

III. Dataset

The dataset used in this research is a question-and-answer dataset utilized in the previous research conducted by Haidir and Purwarianti [17]. This dataset consists of two types of questions: Science and Technology (Saintek) and Social and Humanities (Soshum). The dataset is divided into two sets: a train set consisting of 30 questions and a test set consisting of 6 questions, with a total of 7605 and 1560 rows of answers for each set, respectively. Scoring was conducted by experts in each domain, with scores ranging from 0 to 5. In the following section, this dataset is then referred to as the Saintek-Soshum dataset. The Saintek-Soshum dataset is obtained through NusaCrowd [22], an open-source NLP platform for Indonesian language.

IV. Proposed Method

A. Model Architecture

The architecture of the model used in this research consists of two components: the model body, which is a BERT model used to generate sentence embeddings, and the model head, which is a logistic regression model that produces scores or values based on the features generated by the model body. The model architecture used in this research is shown in Fig 1.

The input to the model is the student's answer and the teacher's answer key, and the output is the predicted score. The input undergoes tokenization before entering the model body, which then generates token embeddings for each token of the input. The model body is structured in a Siamese architecture, meaning that the model body is used to generate

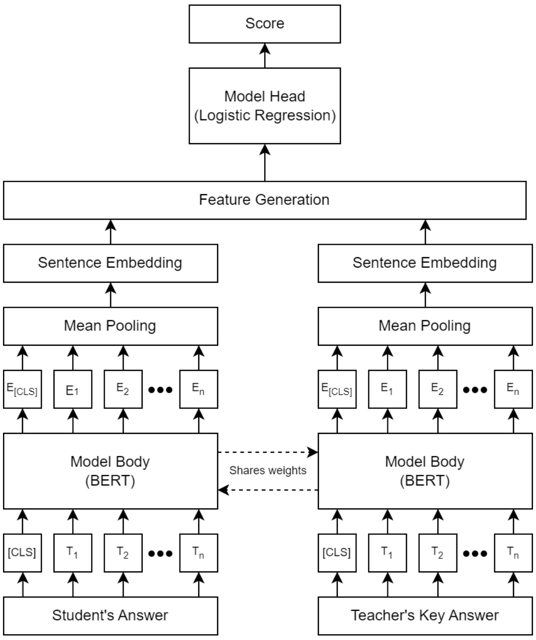


Fig. 1. Proposed Model Architecture

token embeddings for the student's answer and the teacher's answer key is the same model with shared weights.

The token embeddings for each sentence are then fed into the pooling layer with mean pooling, which will average the values of all tokens to produce a sentence embedding. This sentence embedding is further processed in the feature generation component. The output of the feature generation component is a combination of the teacher's key answers embedding, the student's answers embedding, the absolute difference between embedding, and element-wise multiplication between embedding. These features serve as the input to the model head, which ultimately generates the output in the form of a predicted score for the student's answer.

B. Research Workflow

The workflow in this research is divided into three main sections, which are ”Experiment Preparation”, ”Training”, and ”Evaluation”. The overview of this workflow is shown in Fig 2. The first part is ”Experiment Preparation,” which involves determining the methods to be compared and preparing the data. The second part is the ”Training,” which involves the process of training the model using predetermined methods. The third part is the ”Evaluation,” which involves evaluating the performance of the trained model.

Deciding Model to Experiment with

To examine the impact of contrastive learning methods on the performance of models in automated short answer grading systems, this research will compare three models, each employing a different method for training the model body. The first model is referred to as the ”Baseline Model,” which serves as the ground truth. In the baseline model, the BERT model is not finetuned but is used solely to generate sentence embeddings for the model head. The second model is the ”Finetuning Model,” which serves as a comparison to evaluate the performance of contrastive learning against the commonly used finetuning method. In this model, the BERT model will undergo finetuning on the train set. The third model is the ”Contrastive Model,” where the BERT model will be trained using the concept of contrastive learning. All of the three models will use ’bert-base-multilingual’, with the default 768 embedding size.

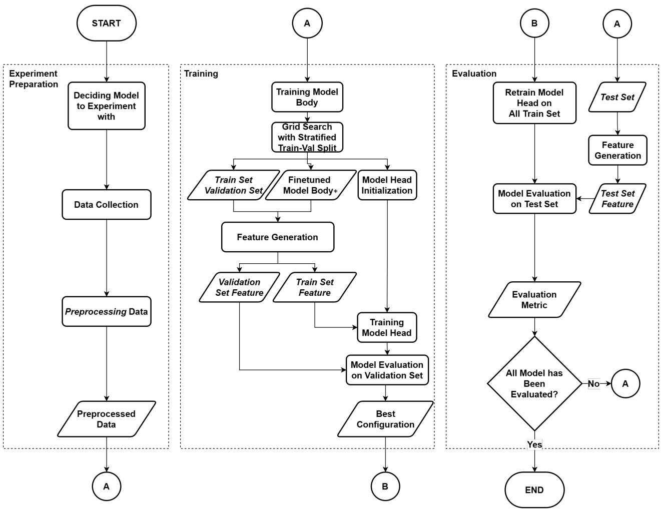


Fig. 2. Workflow Overview - Symbol ∗ denotes that the model is freezed so that no weight is updated

Data Collection

Saintek-Soshum datasets are directly obtained through NusaCrowd, given their open-source nature.

Preprocessing Data

The data preprocessing steps performed on the participant's answers and the answer key in this research are as follows: removal of empty values, removal of punctuation marks, separation of repeated words, removal of escape characters, replacement of words in parentheses, removal of numbers, case-folding, score normalization.

The raw data from the Saintek-Soshum dataset was initially separated into Saintek data and Soshum data. Also, the student's answer was separated from the teacher's answer key. To address this, the datasets are merged into one dataset by combining the Saintek and Soshum data and the student's answer with the teacher's answer key. These preprocessed dataset are shown in Table I

Training Model Body

Each model to be tested (baseline model, finetuning model, and contrastive model) undergoes a slightly different training process. This difference only occurs during the training of the model body, while the training of the model head is the same across all models. Model body are trained with the task to optimize formed embedding through minimizing loss function that is used on each tested model. The finetuning model will use cosine similarity loss, whereas the contrastive model will use contrastive loss.

TABLE I
PREPROCESSED DATASET SAMPLE

index	question_id	response (id—en)	gold_response (id—en)	labels
0	1	bakteri penyubur tanaman — *land-fertilizing bacteria*	bacillus thuringiensis agrobacterium tumefaciens	0.2
1	1	bakteri baik bakteri hasil mutasi — *beneficial bacteria from mutation*	bacillus thuringiensis agrobacterium tumefaciens	0.2
…	…	…	…	…
7434	30	sesuatu yang tetap ada atau tidak punah — *something that continue to exist or not extinct*	tetap seperti keadaan semula — *remains as it was*	1.0
7435	30	ya itu saat alam dalam keaadaan terbaik nya — *nature on its best condition*	tetap seperti keadaan semula — *remains as it was*	0.2

In the baseline model, the model body does not undergo a training process. Instead, the model body is directly used to generate sentence embeddings from the train set and validation set. To replicate a Siamese neural network, the student's answer and the teacher's answer key are inputted separately into the model body, resulting in two sentence embeddings that originate from the same model, parameters, and weights.

In the finetuning model, there is a finetuning process of the model body on the train set. The finetuning process utilizes the following hyperparameter configuration: 1 epoch, learning rate of $2e^{-5}$, batch size of 32, and the loss function utilizing cosine similarity loss.

In the contrastive model, training closely follows the finetuning model, with differences in data format and loss function. The preprocessed training set is adjusted to create contrastive dataset. These contrastive dataset created by labeling rows as 1 for similar answers and 0 for dissimilar ones, based on predefined score ranges. This dataset is then used for finetuning the contrastive model, while other hyperparameters are set the same as the finetuning model.

Grid Search with Cross Validation

Using the Saintek-Soshum train set, 10-hold-out cross-validation is conducted. The separate test set remains untouched during training. The 30-question train set splits into 26 train and 4 validation questions, with criteria of non-overlapping questions. This split maintains a balanced Saintek-Soshum distribution question although among the validation set might have overlapping questions. Question separation is randomized 10 times, without repeated combinations. Throughout grid search and cross-validation, the model body remains frozen to avoid weight updates.

Feature Generation

In each fold iteration of the cross-validation process, the training set and validation set used in that fold will be included in the feature generation process. The resulting features are a combination of sentence embeddings generated by the trained model body for each participant's answer and the answer key. The final feature embedding is a combination of the teacher's key answers embedding, the student's answers embedding, the absolute difference between embedding, and element-wise multiplication between embedding.

Training Model Head

During the model head training, a grid search optimizes two hyperparameters: epochs and learning rate. Each combination is evaluated on the validation set. The objective is predicting and minimizing the MSE loss between actual and predicted scores.

The entire training process of the model head is repeated until the average evaluation metrics for all combinations of hyperparameters are obtained. After obtaining the average evaluation metrics, the best configuration is selected. There are two metrics used in this research, which are MAE and RMSE. However, the selection of the best configuration is based on the configuration that yields the lowest RMSE.

Model Evaluation

The best configuration obtained is then used to retrain the model head. The model head is re-initialized using the best configuration and trained on the entire train set data. Subsequently, the model head is evaluated on the test set, which was not used during the search for the best configuration. The evaluation results on the test set are used as a measure of the overall performance of the model in predicting answer scores. This training and evaluation process is then repeated for all types of models being tested.

C. Contrastive Loss

The contrastive loss is a loss function that maps high-dimensional vector sets I to a low-dimensional space, where similar vectors are brought closer together while dissimilar vectors are pushed apart [23]. The contrastive loss used in this research is an energy-based margin loss. This contrastive loss operates on pairs of vectors $\bar{X}_1, \bar{X}_2 \in I$ with a label Y. In the implementation, there was a slight adjustment from the [23], where $Y = 1$ indicates a similar pair of vectors, and $Y = 0$ indicates a dissimilar pair of vectors. The contrastive loss is formulated in Equation 1:

$$\mathcal{L} = (1-Y)\frac{1}{2}\{max(0, m - D_W)\}^2 + (Y)\frac{1}{2}(D_W)^2 \quad (1)$$

where D_W represents the distance between the vector pairs $\bar{X}_1$ and $\bar{X}_2$ in the embedding space, and m is a margin that controls the dissimilarity threshold. The contrastive loss encourages similar pairs to have a small distance (D_W) and dissimilar pairs to have a distance larger than the margin (m), which used the default value of 0.5.

V. Experimental Result

A. Score Thresholds Effect on Model Performance

To demonstrate the impact of score thresholds on contrasive model performance, this study tested different limits for positive and negative rows using grid search and cross-validation. Same number within square brackets implies an data with exact score match with the threshold, while two distinct numbers indicate inclusion of scores within that range. Data outside the threshold will be excluded during training. The validation set's evaluation outcomes for the MAE and RMSE metrics are shown in Table II and Table III, respectively.

TABLE II
MAE Result on Various Score Limit Combination for Contrastive Model

Negative Limit	Positive Limit				
	[1.0, 1.0]	[0.8, 0.8]	[0.8, 1.0]	[0.6, 1.0]	[0.6, 0.8]
[0.0, 0.0]	0.2497	0.2322	0.2362	0.2356	0.2477
[0.2, 0.2]	0.1717	0.1556	0.1644	0.1628	0.1957
[0.0, 0.2]	0.1643	0.1528	**0.1524**	0.1599	0.2045
[0.0, 0.4]	0.1732	0.1610	0.1638	0.1547	0.1762
[0.2, 0.4]	0.1858	0.1728	0.1755	0.1733	0.1881

TABLE III
RMSE Result on Various Score Thresholds Combination for Contrastive Model

Negative Limit	Positive Limit				
	[1.0, 1.0]	[0.8, 0.8]	[0.8, 1.0]	[0.6, 1.0]	[0.6, 0.8]
[0.0, 0.0]	0.2860	0.2689	0.2736	0.2735	0.2848
[0.2, 0.2]	0.2046	0.1933	0.2004	0.2022	0.2340
[0.0, 0.2]	0.1979	0.1927	**0.1901**	0.1970	0.2426
[0.0, 0.4]	0.2058	0.1986	0.2005	0.1915	0.2167
[0.2, 0.4]	0.2196	0.2123	0.2137	0.2127	0.2332

Based on the grid search results for score thresholds, the highest performance of the model is achieved with an MAE of 0.1524 and an RMSE of 0.1901. These results are achieved using the optimal score thresholds for the Saintek-Soshum dataset are $[0.8, 1.0]$ for positive rows and $[0.0, 0.2]$ for negative rows. The grid search also indicates that the determination of score thresholds significantly impacts the model's performance.

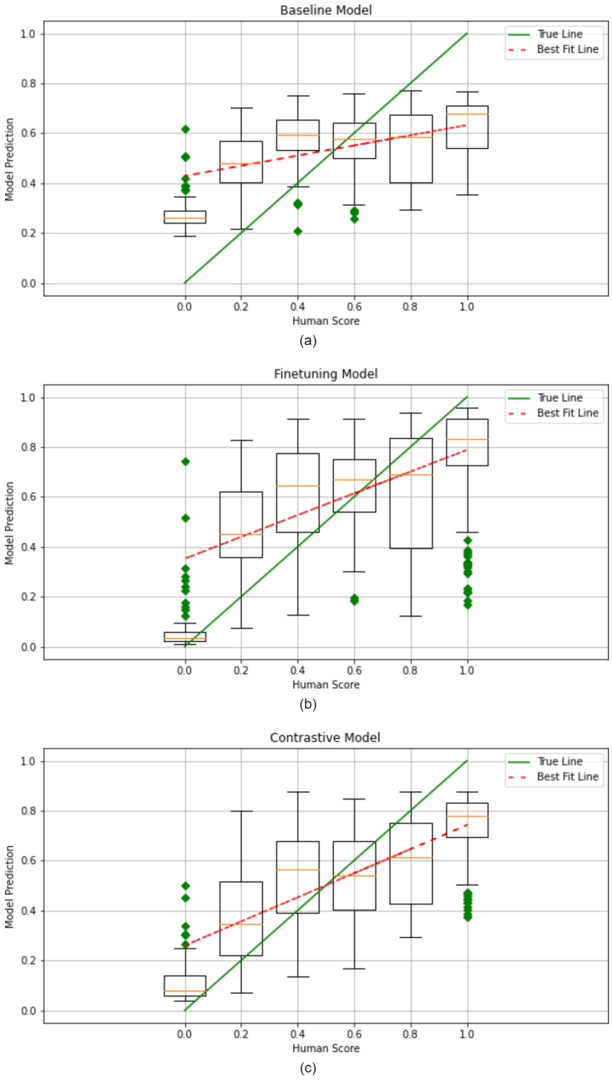
B. Model Comparison

The comparison between the models was conducted by evaluating each model with their respective best hyperparameter configurations on the test set. The evaluation results of each model, along with the best hyperparameter configurations used, are shown in Table IV. The contrastive model excels in prediction, as evident from lower losses across all measured metrics. It outperforms the baseline and finetuning models by reducing MAE by 0.053 and 0.021, and RMSE by 0.050 and 0.037, respectively. Furthermore, all our models performed better than the one reported in [17], which is similar to the baseline model. The key difference is in how we pool output vectors from BERT: the baseline model used simple average pooling, while [17] used WK-pooling.

TABLE IV
Model Performance Result

Model	*Hyperparameter*				MAE	RMSE
	Epoch Head	LR Head	Positive Limit	Negative Limit		
[17]	15	$1e^{-3}$	-	-	0.288	0.378
Baseline	64	$1e^{-4}$	-	-	0.244	0.281
Finetuning	128	$1e^{-3}$	-	-	0.212	0.268
Contrastive	256	$1e^{-4}$	$[0.8, 1.0]$	$[0.0, 0.2]$	**0.191**	**0.231**

An analysis was then conducted on the prediction results of each tested model. The visualization of the prediction results is shown in Fig. 3 which focuses on the prediction results within each score range.



(a)

(b)

(c)

Fig. 3. Box Plot of Each Model Prediction Result Compared to Human Scoring

Fig. 3 shows that the baseline model predicts scores mostly between 0.5 to 0.7, implying undifferentiated embeddings, while the finetuning and contrastive models create more distinct embeddings within each score range. However, the finetuning model struggles with predicting scores of 0.8. Overall, the contrastive model's more centralized predictions yield a smaller mean error compared to the finetuning model.

VI. Conclusion

In this study, experiments were conducted to assess the performance of a transformer-based model using contrastive learning in the task of automated short answer assessment. The contrastive model's performance was benchmarked against two alternatives: a baseline using pretrained BERT, and a finetuned BERT using cosine similarity.

In addition to the primary objective, an analysis of the impact of score thresholds on the performance of the contrastive model demonstrated a direct relationship between score thresholds and model performance, emphasizing the importance of determining an optimal threshold to achieve the best model performance.

The main objective was accomplished by comparing the performance of each model. Contrastive model achieved a reduction in prediction errors, with a decrease in mean absolute error (MAE) of 21.72% and 9.90% and a decrease in root mean squared error (RMSE) of 17.79% and 13.80%, compared to the baseline and finetuning models, respectively. The contrastive model achieved a performance metric of 0.191 for MAE and 0.231 for RMSE.

The examined contrastive model enhances performance over baselines, although sensitive to score thresholds. This research result opens up the potential for further development utilizing contrastive learning. Furthermore, the contrastive learning method used in this research is a basic approach, leaving room for more advanced techniques to be explored.

Concerning the dataset, Saintek-Soshum is evaluated by 7 experts, resulting in reduced bias. The various participant responses are also go beyond simply copying online content, enriching information acquisition. Such factors hold significance in the collection of good Question-Answer datasets.

References

[1] A. Baylari and G. Montazer, "Design a personalized e-learning system based on item response theory and artificial neural network approach," *Expert Systems with Applications*, vol. 36, no. 4, pp. 8013–8021, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741740800777X

[2] E. Sutanta, "Konsep dan implementasi e-learning," *Yogyakarta: IST Akprind*, pp. 10–12, 2009.

[3] S. Naseer, "Perspective chapter: Advantages and disadvantages of online learning courses," pp. 1–11, 07 2023.

[4] M. A. Almaiah, A. Al-Khasawneh, and A. Althunibat, "Exploring the critical challenges and factors influencing the e-learning system usage during covid-19 pandemic," *Education and information technologies*, vol. 25, no. 6, pp. 5261–5280, 2020.

[5] S. Syahrir, Y. Supriyati, and A. Fauzi, "Evaluasi dampak program pendidikan jarak jauh (pjj) melalui model cipp pada kinerja dosen aspek pembelajaran pada masa pendemi covid 19," *Jurnal Ilmiah Mandala Education*, vol. 7, no. 1, 2021.

[6] R. Wijaya, M. Lukman, and D. Yadewani, "Dampak pandemi covid19 terhadap pemanfaatan e learning," *Jurnal Dimensi*, vol. 9, no. 2, pp. 307–322, 2020.

[7] S. S. Kusumawardani and S. A. I. Alfarozi, "Transformer encoder model for sequential prediction of student performance based on their log activities," *IEEE Access*, vol. 11, pp. 18 960–18 971, 2023.

[8] M. A. Fauzi, D. C. Utomo, B. D. Setiawan, and E. S. Pramukantoro, "Automatic essay scoring system using n-gram and cosine similarity for gamification based e-learning," in *Proceedings of the International Conference on Advances in Image Processing*, ser. ICAIP '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 151–155. [Online]. Available: https://doi.org/10.1145/3133264.3133303

[9] b. Shen, N. McCaughtry, J. Martin, A. Garn, N. Kulik, and M. Fahlman, "The relationship between teacher burnout and student motivation," *British Journal of Educational Psychology*, vol. 85, 07 2015.

[10] A. K. Arens and A. J. Morin, "Relations between teachers' emotional exhaustion and students' educational outcomes." *Journal of Educational Psychology*, vol. 108, no. 6, p. 800, 2016.

[11] M. Beseiso, O. A. Alzubi, and H. Rashaideh, "A novel automated essay scoring approach for reliable higher educational assessments," *Journal of Computing in Higher Education*, vol. 33, no. 3, pp. 727–746, 2021.

[12] Y. Zhang, R. Tang, and J. J. Lin, "Explicit pairwise word interaction modeling improves pretrained transformers for english semantic similarity tasks," *ArXiv*, vol. abs/1911.02847, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:207848032

[13] R. Fitri and A. N. Asyikin, "Aplikasi penilaian ujian essay otomatis menggunakan metode cosine similarity," *Poros Teknik*, vol. 7, no. 2, pp. 88–94, 2015.

[14] U. Hasanah and D. A. Mutiara, "Perbandingan metode cosine similarity dan jaccard similarity untuk penilaian otomatis jawaban pendek," in *SENSITIf: Seminar Nasional Sistem Informasi dan Teknologi Informasi*, 2019, pp. 1255–1263.

[15] A. A. P. Ratna, H. Khairunissa, A. Kaltsum, I. Ibrahim, and P. D. Purnamasari, "Automatic essay grading for bahasa indonesia with support vector machine and latent semantic analysis," in *2019 International Conference on Electrical Engineering and Computer Science (ICECOS)*. IEEE, 2019, pp. 363–367.

[16] 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," in *MATEC web of conferences*, vol. 164. EDP Sciences, 2018, p. 01037.

[17] M. H. Haidir and A. Purwarianti, "Short answer grading using contextual word embedding and linear regression," *Jurnal Linguistik Komputasional*, vol. 3, no. 2, pp. 54–61, 2020.

[18] H. R. Salim, C. De, N. D. Pratamaputra, and D. Suhartono, "Indonesian automatic short answer grading system," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 3, pp. 1586–1603, 2022.

[19] R. A. Rajagede, "Improving automatic essay scoring for indonesian language using simpler model and richer feature," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, pp. 11–18, 2021.

[20] N. Peinelt, D. Nguyen, and M. Liakata, "tBERT: Topic models and BERT joining forces for semantic similarity detection," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 7047–7055. [Online]. Available: https://aclanthology.org/2020.acl-main.630

[21] F. W. Mutinda, S. Yada, S. Wakamiya, and E. Aramaki, "Semantic textual similarity in japanese clinical domain texts using bert," *Methods of Information in Medicine*, vol. 60, no. S 01, pp. e56–e64, 2021.

[22] S. Cahyawijaya, H. Lovenia, A. F. Aji, G. I. Winata, B. Wilie, R. Mahendra, C. Wibisono, A. Romadhony, K. Vincentio, F. Koto, J. Santoso, D. Moeljadi, C. Wirawan, F. Hudi, I. H. Parmonangan, I. Alfina, M. S. Wicaksono, I. F. Putra, S. Rahmadani, Y. Oenang, A. A. Septiandri, J. Jaya, K. D. Dhole, A. A. Suryani, R. A. Putri, D. Su, K. Stevens, M. N. Nityasya, M. F. Adilazuarda, R. Ignatius, R. Diandaru, T. Yu, V. Ghifari, W. Dai, Y. Xu, D. Damapuspita, C. Tho, I. M. K. Karo, T. N. Fatyanosa, Z. Ji, P. Fung, G. Neubig, T. Baldwin, S. Ruder, H. Sujaini, S. Sakti, and A. Purwarianti, "Nusacrowd: Open source initiative for indonesian nlp resources," 2022.

[23] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1735–1742.