Artificial Intelligence Systems and Its Applications (AISA)
Volume 1, Issue 2, December 2025
http://journals.cognispectra.com/index.php/aisa/index
E-ISSN: 3110-0457

Text Representation Method Analysis and Its Implementation in Automatic Essay Scoring System

Alya Zakhira Anjani[1], Divi Galih Prasetyo Putri[1], Widhy Hayuhardhika Nugraha Putra[2]

[1] Universitas Gadjah Mada, Sleman, Indonesia

[2] Universiti Tun Hussein Onn Malaysia, Batu Pahat, Malaysia

Abstract: The automatic essay scoring system is one of many problems in terms of natural language processing (NLP) that has long been studied. In an automatic essay scoring system, there is an approach using text similarity with cosine similarity method to determine correct and incorrect predictions. However, the text representation phase is also an important phase. This study compares the performance of three text representation methods in their implementation into an automatic essay scoring system. The methods are Indonesian Version of Bidirectional Encoder from Transformers (IndoBERT), Embeddings from Language Model (ELMo), and FastText. In addition, the combination of each method with WordNet as an additional lexical resource is also compared. The result of comparison using dataset "Indonesian Query Answering Dataset for Online Essay Test System" shows that the combination of IndoBERT and WordNet model has the best performance proven with highest accuracy achieved being 0.69, precision being 0.54, recall being 0.81, and F1-score being 0.48. Then the model was implemented as an essay evaluation feature development for the Certified Government Accounting Associate (CGAA) Exam Simulation site. The feature performance test results show an average load time of 418.8 ms when accessed by 10 users simultaneously and 15064 ms when accessed by 100 users simultaneously. The features developed are expected to be able to support the evaluation process more efficiently.

Keywords: Text Representation Model, IndoBERT, ELMo, FastText

Article History:

Received: 22 July 2025 Accepted: 28 December 2025 Published: 31 December 2025

Corresponding Author: Alya Zakhira Anjani, Email: alya.zak2002@mail.ugm.ac.id

DOI: 10.65917/aisa.v1i2.30

1 Introduction

The rapid advancement of technology has significantly influenced various existing systems, enhancing both their efficiency and accuracy. One of the most impacted sectors is education, where technological innovations have transformed traditional practices [2]. A critical component of education is the assessment of student competencies, which is still predominantly conducted manually by teachers and lecturers. However, as the ratio of students to educators increases, manual assessment becomes increasingly inefficient and time-consuming. Moreover, manual grading is prone to human error, leading to potential inaccuracies [3]. This issue is especially evident in essay-based and short-answer evaluations, which require extensive time and are inherently subjective due to their reliance on human judgment. To address these challenges, the implementation of automated grading systems offers a more objective and efficient alternative [4].

Automated grading can be achieved through machine learning, particularly within the domain of Natural Language Processing (NLP). One widely used NLP technique for such systems is text similarity, which evaluates student responses by comparing them with predefined reference answers [5]. Among various similarity measurement methods, Cosine Similarity is commonly employed. In general, NLP tasks involve two key stages: transforming raw text into numerical data (such as matrices or vectors), and then designing models to process this representation [6]. Numerous text representation algorithms exist, with their performance varying by context. Consequently, it is necessary to analyze and select the most appropriate algorithm for a given system, such as an automated grading system.

Based on the review by Putnikovic and Jovanovic [7], text representation techniques, particularly text embedding, can be categorized into four types: word embedding, contextual embedding, sentence embedding, and sense embedding. This study compares word and contextual embeddings, two widely adopted but contrasting approaches. Specifically, it investigates IndoBERT and ELMo as representatives of contextual embedding, and FastText as a representative of word embedding. While word embeddings are computationally lighter, they often fail to capture semantic context effectively. To mitigate this limitation, each model in this study is enhanced with additional lexical resources, namely WordNet, to better represent semantic meaning.

Previous studies have explored model combinations. For example, Alobed et al. [16] integrated TF-IDF with WordNet, employing similarity measures such as Euclidean, Jaccard, and Cosine in an automated grading system. The combined model showed improvements in scoring accuracy. Additionally, a more complex hybrid model involving BERT, LSTM, and CNN was proposed by Kaya and Cicekli [12], demonstrating strong performance in classifying correct and incorrect answers.

This study applies the models to a case study involving a simulation of the Certified Government Accounting Associate (CGAA) exam, a certification offered by the Indonesian Institute of Accountants (IAI). According to the official IAI website, the CGAA certification assesses an individual's ability to

Artificial Intelligence Systems and Its Applications (AISA)
Volume 1, Issue 2, December 2025
http://journals.cognispectra.com/index.php/aisa/index
E-ISSN: 3110-0457

prepare financial statements for central and regional governments [8]. A simulation platform for the CGAA exam has been previously developed as part of a research initiative to prepare students, particularly those in the Public Sector Accounting program at Universitas Gadjah Mada. This platform, accessible via desktop and mobile devices, has been in use since October 2022. Despite its usefulness, the system's evaluation functionality is limited, as the essay grading component requires manual comparison with reference solutions.

To improve this, this study proposes an enhanced evaluation feature that includes automated prediction of whether a student's answer is correct or incorrect. The model determined to be the most effective through the earlier comparison will be implemented as a new Application Programming Interface (API) within the platform. The integration will be evaluated through user acceptance testing, integration testing to ensure correct input-output behavior, and performance testing using Apache JMeter to assess system efficiency and scalability.

2 Related Work

Research in the domain of Automatic Short Answer Grading (ASAG) has evolved considerably, particularly with the integration of deep learning and text similarity algorithms. Various approaches have been proposed to enhance the accuracy and objectivity of automated grading systems.

Salim et al. [9] developed an ASAG system in the Indonesian language, comparing the performance of BERT-based models and Ridge Regression. Their experiments involving eight BERT variants (including six BERT and two ALBERT models) demonstrated that transformer-based models significantly outperformed Ridge Regression, with the best model (ALBERT lite-base-P2) achieving a Pearson correlation of 0.9508 and RMSE of 0.4138. This highlights the effectiveness of contextual embeddings in ASAG, although their evaluation lacked real-time system integration or hybrid modeling approaches.

Zhang et al.[10] introduced an LSTM-based model for semi-open-ended question grading in reading comprehension tasks. Their model utilized both labeled student responses and contextual knowledge from Wikipedia. The combination of general and domain-specific information yielded a QWKappa score of 0.6243, outperforming traditional classifiers such as Logistic Regression and Naive Bayes. This study emphasizes the advantage of hybrid features for capturing semantic relevance.

In a study focused on text similarity, Susanto et al. [11] evaluated five similarity algorithms, including Cosine, Jaccard, TF-IDF variants, and LSA using 371 manually scored short answers. They found that TF-IDF with Jaccard similarity yielded the lowest RMSE (0.3921), suggesting its relative effectiveness

in capturing lexical overlap. However, while LSA produced a higher RMSE (0.5368), it more closely mimicked human grading patterns, albeit with greater variability.

Kaya and Cicekli [12] explored a novel hybrid architecture integrating BERT, LSTM, and CNN. Their model achieved over 80% accuracy for binary classification and 76–73% for three-class classification on benchmark datasets such as SciEntsBank and Beetle. Moreover, on the Mohler dataset, it reached a Pearson correlation of 0.747 with human scores. These findings confirm the robustness of deep hybrid models in ASAG, particularly when addressing nuanced answer scoring.

Wijaya [13] investigated BERT for Indonesian ASAG tasks, yielding strong results: Cohen's Kappa of 0.75, precision of 0.94, and F1-score of 0.95 across 48 high school answers. This study validated the feasibility of using pretrained transformers for Indonesian education contexts but did not explore model extensibility or hybridization.

Balaha and Saafan [14] compared text representation algorithms Word2Vec, FastText, GloVe, and USE across tools like Gensim, SpaCy, and NLTK, using datasets such as Quora Question Pairs and UNT Short Answers. USE Large (SpaCy) achieved the highest accuracy (77.95%), while FastText (SpaCy/Gensim) showed the lowest RMSE (1.09). This comparative analysis underscores that model performance depends not only on the algorithm but also on its implementation framework.

Patil et al. [6] conducted a comprehensive review of text representation algorithms, tracing their evolution from basic pattern-matching to neural-based models. They classified them into static embeddings (e.g., Word2Vec, FastText, GloVe) and dynamic/contextual embeddings (e.g., ELMo, BERT, GPT). Their synthesis shows that neural embeddings consistently outperform older methods in preserving semantic and contextual meaning.

In a similar direction, Hendre et al. [15] compared six embedding strategies (TF-IDF, Jaccard, GloVe, ELMo, and two versions of Google's Sentence Encoder) for essay evaluation. Their results showed that GSE-Large and ELMo achieved the highest d-prime scores (2.8375 and 2.1527, respectively), indicating superior discrimination power. This supports the use of advanced contextual embeddings in enhancing grading accuracy.

Putnikovic and Jovanovic [7] performed a systematic review of 17 studies on embedding algorithms in ASAG. They found a predominance of word embeddings combined with Cosine Similarity, but also highlighted the emergence of contextual, sentence, and sense embeddings. Notably, BERT and its variants (e.g., RoBERTa, ALBERT, DistilBERT) were frequently used due to their strong contextualization capabilities. Their work underlines the importance of matching the embedding strategy to the target application.

In a study relevant to this paper, Alobed et al. [16] designed an ASAG system for Arabic essays using a combination of WordNet and text similarity algorithms. They found that Cosine Similarity combined with WordNet achieved the best performance (Pearson correlation of 0.97), significantly outperforming

algorithms without lexical support. This highlights the effectiveness of enhancing traditional models with lexical resources for better context handling.

A review of prior studies reveals that Cosine Similarity is one of the most widely implemented methods for measuring text similarity in automated grading systems. Accordingly, this study adopts Cosine Similarity as its similarity measurement method. Additionally, previous research highlights the strong performance of BERT-based models for text representation, with ELMo also emerging as a powerful contextual embedding method, and FastText frequently used as a representative of word embeddings due to its efficiency.

While these individual models have been studied separately, there remains limited comparative analysis involving IndoBERT, ELMo, and FastText, particularly when combined with lexical resources such as WordNet. This study addresses that gap by conducting a systematic comparison of these three embedding approaches and evaluating the impact of WordNet integration on their performance.

Unlike most existing works, this study also focuses on real-world implementation by integrating the best-performing model into the CGAA exam simulation platform as a new evaluation feature. Furthermore, the comparison is conducted using the Indonesian Query Answering Dataset for Online Essay Test System, a dataset that has not been utilized in previous comparative research of this kind. This dual contribution, both in terms of empirical model comparison and practical deployment, distinguishes this study from prior work in the field.

3 Methods

3.1 Research design

This study adopts an experimental research design to compare the performance of text representation methods: IndoBERT, ELMo, FastText and its combination with WordNet in an automated short essay grading system. The goal of the experiment is to evaluate the accuracy of automated grading in comparison to human judgment.

3.2 Data source

The dataset used in this study is the “Indonesian Query Answering Dataset for Online Essay Test System”, developed by Rahutomo and Roshinta [1]. The dataset contains a total of 2,162 entries from four topics: politics, sports, lifestyle, and technology. Each entry consists of a question, student’s answer, reference (key) answer, and both manual and automatic scores.

Table 1 illustrates the structure of the dataset before preprocessing. For the purpose of binary classification, a new label column was added to the dataset, categorizing each answer as either correct (1) or incorrect (0). This label was derived by converting the average manual score into a binary value

based on a predefined threshold. Additionally, irrelevant columns were removed to focus the analysis on the comparison of similarity algorithms.

Table 1: Raw dataset.

No	Siswa	Jawaban	Manual 1	Manual 2	Manual 3	Cos	Euc	Jac	Cos Stemm	Euc Stemm	Jacc Stemm	Rata Manual	Error Cosine	Error Euclidean	Error Jaccard	Error Cos Stem	Error Euc Stem	Error Jacc Stem
1	siswa_2	sumber tenaga, pemanis alami, menjaga sistem imun, dan sebagai keseimbangan tubuh	20	35	25	8	72	4	8	72	4	26.66666667	70	170	85	70	170	85

Table 2 shows an example of the dataset after preprocessing (Preprocessed dataset)

question_id	student_answer	key_answer	manual_average_score	label
lifestyle_1	sumber tenaga, pemanis alami, menjaga sistem imun, dan sebagai keseimbangan tubuh	Fungsi karbohidrat adalah sebagai pemasok energi, dapat memperlancar proses pada pencernaan, memberikan efek kenyang dengan kandungan selulosa-nya dan penyeimbang asam dan basa dalam tubuh	26.66666667	0

Table 3 shows the distribution of correct and incorrect labels after data preprocessing (Preprocessed dataset)

Topics	Correct Label	Wrong Label
Lifestyle	15	553
Sport	99	445
Politics	186	349
Technology	104	411

3.3 Preprocessing techniques

The first step was to convert the manual scores of student answers into binary labels. This conversion was based on the study by Lan [17], which used a similarity threshold of 0.7 on a scale from 0 to 1 to determine correct answers. In contrast, the dataset used in this study includes three manual scores per student answer, each rated on a 0–100 scale by different human assessors. To align with Lan's approach, the average of the three manual scores was calculated, and a threshold of 70 was applied,

corresponding to a 0.7 normalized value. Answers with an average score above 70 were labeled as correct (1), while those with scores equal to or below 70 were labeled as incorrect (0).

The second preprocessing step involved cleaning the dataset by removing any rows with missing values. Additionally, student answers that contained only a single character were excluded to reduce noise and potential errors during the text similarity processing. This cleaning step helps ensure data quality and more reliable model evaluation. The third preprocessing step involved text normalization to ensure more accurate text representation. This step included several processes: removing punctuation, case folding (converting all text to lowercase), and whitespace correction, such as removing extra spaces or newline characters.

3.4 Algorithm and Models

This study compares the performance of three text representation models, and further examines the effect of combining each model with WordNet for synonym expansion. In the implementation phase, each model will represent two types of text: (1) the original text, and (2) the WordNet-augmented text that includes synonym expansions.

The synonym expansion process is based on the work of Barbouch et al. [18], which incorporates both external and internal WordNet integration. Externally, synonyms are generated by identifying the Part-of-Speech (POS) tag of each word and expanding only those tagged as nouns, verbs, adjectives, or adverbs. Internally, the original and expanded texts are embedded separately, and the resulting vectors are combined into a single unified vector representing the text. This expansion process is performed before the embedding stage, so models that utilize WordNet operate on augmented text.

The first model used is IndoBERT, a pre-trained BERT model for the Indonesian language. Before being fed into the model, text is processed by IndoBERT's custom tokenizer, which performs tokenization and adds special tokens: [CLS] at the beginning and [SEP] as a separator. IndoBERT generates three types of embeddings:

- Token Embedding: Represents each word based on the model's trained vocabulary.
- Token-Type Embedding: Indicates sentence segment (either 0 or 1).
- Positional Embedding: Marks the position of each token in the input sequence. IndoBERT supports sequences up to 512 tokens.

After initial embedding, the data is passed through 12 encoder layers. Each encoder consists of a Multi-Head Attention mechanism and a Feedforward Neural Network (FFNN). The attention mechanism evaluates the importance of each word relative to its context, while the FFNN processes the output

further to be passed to the next layer. The final output is a normalized vector representing the semantic features of the input text. This is used as the final text representation vector from IndoBERT.

The second model is ELMo (Embeddings from Language Models), which generates a 1024-dimensional vector for each input text. Unlike static embeddings, ELMo produces context-sensitive embeddings, meaning that word representations vary depending on their surrounding words. The third model is FastText, which represents text as a 300-dimensional vector. FastText differs from word-based embeddings by representing each word as a combination of character n-grams, which allows it to handle rare or misspelled words more effectively. Each of these models, IndoBERT, ELMo, and FastText, will be tested in both standalone and WordNet-augmented configurations.

3.5 Evaluation metrics

Before evaluation, the text similarity between student answers and reference answers is calculated using the Cosine Similarity method. This results in a similarity score ranging from 0 to 1 for each answer pair. Equation 1 shows the cosine similarity method.

$$similarity = \frac{A \cdot B}{||A||\,||B||} \quad (1)$$

To enable binary classification and evaluation, these decimal similarity scores are then thresholded. Following the approach used in prior research, a threshold of 0.7 is applied based on study by Lan [17]. Scores greater than 0.7 are classified as 1 (correct) and scores equal to or less than 0.7 are classified as 0 (incorrect). The evaluation is then conducted using a Confusion Matrix, which compares the predicted labels (based on similarity scores) with the actual labels derived from manual scoring. These labels have four categories such as true positive (TP), true negative (TN), false positive (FP), and false negative (FN). [13]

- True Negative (TN)

  A true negative result means that the model predicted a data point as belonging to the negative category, and the actual data also belongs to the negative category.
- True Positive (TP)

  A true positive result means that the model predicted a data point as belonging to the positive category, and the actual data also belongs to the positive category.
- False Negative (FN)

  A false negative result means that the model predicted a data point as belonging to the negative category, but the actual data belongs to the positive category.
- False Positive (FP)

  A false positive result means that the model predicted a data point as belonging to the positive category, but the actual data belongs to the negative category.

Artificial Intelligence Systems and Its Applications (AISA)
Volume 1, Issue 2, December 2025
http://journals.cognispectra.com/index.php/aisa/index
E-ISSN: 3110-0457

The confusion matrix enables calculation of key performance metrics such as accuracy, precision, recall, and F1-score. These metrics are calculated based on Equation 2, 3, 4, and 5.

- Accuracy: The proportion of correctly classified instances

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \quad (2)$$

- Precision: The ratio of true positives to all predicted positives

$$Precision = \frac{TP}{(TP+FP)} \quad (3)$$

- Recall: The ratio of true positives to all actual positives

$$Recall = \frac{TP}{(TP+FN)} \quad (4)$$

- F1-Score: The harmonic mean of precision and recall

$$F1\ Score = \frac{(2 \times Recall \times Precision)}{(Recall + Precision)} \quad (5)$$

3.5 Implementation

After identifying the most suitable algorithm, the selected model was implemented (deployed) in the back-end of the CGAA exam simulation system using an API (Application Programming Interface). The implementation was carried out using the Flask framework in Python. Following the deployment, the front-end of the system was updated to enable access to the API over the network.

3.5 Load Testing

Load testing was conducted using the Apache JMeter tool to evaluate the API's behavior under simulated loads of 10 and 100 concurrent users. The test results included total response time in milliseconds (ms), processing time per request, and the error rate or percentage of failed requests. These results were used as a basis to assess whether the API implementation is sufficiently effective and meets performance expectations.

4 Results and Discussion

4.1 Result

All text similarity measurement results were evaluated against the ground truth labels in the dataset. The evaluation was carried out using a confusion matrix to calculate the values of True Negative (TN), False Positive (FP), False Negative (FN), and True Positive (TP). Table 1 presents the results of the evaluation using the confusion matrix.

Table 1: Confusion Matrix Result

Topics	Model	TN	FP	FN	TP
Lifestyle	IndoBERT	369	182	0	15
	ELMo	40	511	0	15
	FastText	72	479	0	15
	IndoBERT and WordNet	379	172	1	14
	ELMo and WordNet	29	522	0	15
	FastText and WordNet	68	483	0	15
Sport	IndoBERT	181	264	1	98
	ELMo	32	413	0	99
	FastText	77	368	0	99
	IndoBERT and WordNet	195	250	2	97
	ELMo and WordNet	19	426	0	99
	FastText and WordNet	61	384	0	99
Politics	IndoBERT	164	184	56	130
	ELMo	9	339	49	137
	FastText	58	290	118	68
	IndoBERT and WordNet	194	154	55	131
	ELMo and WordNet	5	343	49	137
	FastText and WordNet	128	220	131	55
Technology	IndoBERT	167	242	4	100
	ELMo	41	368	3	101
	FastText	39	370	2	102
	IndoBERT and WordNet	190	219	4	100
	ELMo and WordNet	24	385	3	101
	FastText and WordNet	37	372	2	102

From these values, several key performance metrics were derived: accuracy, precision, recall, and F1-score. The overall performance results obtained using these evaluation metrics are summarized in Table 2. These metrics provide a comprehensive understanding of the model's ability to classify student answers correctly based on the similarity scores.

Artificial Intelligence Systems and Its Applications (AISA)
Volume 1, Issue 2, December 2025
http://journals.cognispectra.com/index.php/aisa/index
E-ISSN: 3110-0457

Table 2: Performance Metrics Result

Topics	Model	*Accuracy*	*Precision*	*Recall*	F1-*score*
Lifestyle	IndoBERT	0,68	0,54	0,83	0,47
	ELMo	0,10	0,51	0,54	0,10
	FastText	0,15	0,52	0,57	0,15
	IndoBERT and WordNet	0,69	0,54	0,81	0,48
	ELMo and WordNet	0,08	0,51	0,53	0,08
	FastText and WordNet	0,15	0,52	0,56	0,14
Sport	IndoBERT	0,51	0,63	0,70	0,50
	ELMo	0,24	0,60	0,54	0,23
	FastText	0,32	0,61	0,59	0,32
	IndoBERT and WordNet	0,54	0,63	0,71	0,52
	ELMo and WordNet	0,22	0,59	0,52	0,20
	FastText and WordNet	0,29	0,60	0,57	0,29
Politics	IndoBERT	0,55	0,58	0,59	0,55
	ELMo	0,27	0,22	0,38	0,23
	FastText	0,24	0,26	0,27	0,24
	IndoBERT and WordNet	0,61	0,62	0,63	0,60
	ELMo and WordNet	0,27	0,19	0,38	0,22
	FastText and WordNet	0,34	0,35	0,33	0,33
Technology	IndoBERT	0,52	0,63	0,68	0,51
	ELMo	0,28	0,57	0,54	0,27
	FastText	0,27	0,58	0,54	0,26
	IndoBERT and WordNet	0,57	0,65	0,71	0,55
	ELMo and WordNet	0,24	0,55	0,51	0,23
	FastText and WordNet	0,27	0,58	0,54	0,26

Subsequently, load testing was conducted using the Apache JMeter application. For comparison purposes, testing was performed on both the IndoBERT implementation without WordNet and the IndoBERT implementation with WordNet. Therefore, the testing was divided into four stages: two stages for the IndoBERT implementation without WordNet, and two stages for the IndoBERT

implementation with WordNet. The first two stages involved testing with 10 virtual users, while the remaining two stages involved testing with 100 virtual users. Based on the results, it can be concluded that the implementation of the IndoBERT and WordNet combination model improves accuracy; however, it also increases response time compared to the IndoBERT-only implementation. The longer processing time in the combined model is due to the additional text expansion process, which is not present in the IndoBERT-only model. A detailed comparison of the performance between the two models is presented in Table 3.

Table 3: Load Testing Result

Model	Users	Average time (ms)	Minimum time (ms)	Maximum time (ms)
IndoBERT	10	178,2	95	473
IndoBERT and WordNet	10	418,8	125	581
IndoBERT	100	8806	6014	9822
IndoBERT and WordNet	100	15064	10840	17037

4.1 Discussion

Based on the evaluation results of all models, it can be concluded that IndoBERT combined with WordNet achieved the highest accuracy in representing text for the text similarity task. Therefore, this model was selected for deployment in the next stage using a Flask-based API system. However, despite IndoBERT's superior performance, the accuracy of all six tested configurations (three models with and without WordNet) remained below 0.7. This highlights the complexity of the task and the challenges in achieving high precision in automated short-answer scoring. The integration of WordNet with IndoBERT led to a notable improvement in accuracy, suggesting that synonym expansion using WordNet enhances the model’s ability to capture contextual meaning. A similar, though more limited, improvement was observed in the combination of FastText and WordNet model, particularly within one specific topic. When averaged across all topics, FastText still performed better when combined with WordNet than when used alone.

Overall, IndoBERT demonstrated the best performance and highest accuracy compared to the other models. This may be attributed to IndoBERT’s contextual understanding of language, enabled by its transformer architecture. Moreover, IndoBERT has been pre-trained specifically on Indonesian text, which likely enhances its ability to process Indonesian language more effectively than general-purpose models. In contrast, the ELMo model recorded the lowest accuracy among the three. One contributing factor is that ELMo lacks an Indonesian-specific version, unlike IndoBERT and FastText. A study by

Kurniawan et al. [19] also indicated that, as of the time of their research, there had been no implementation of ELMo specifically for the Indonesian language. Their results showed that using ELMo in POS tagging tasks underperformed compared to IndoBERT-based approaches. The FastText model, although slightly outperforming ELMo, also exhibited relatively low accuracy. This is likely because FastText is a word-level embedding model that does not take sentence-level context into account. As a result, it may struggle with understanding nuanced textual meaning in short-answer settings.

An analysis of the confusion matrix revealed that both ELMo and FastText share certain limitations. In particular, they generated a high number of false positives. This implies that both models tended to produce inflated similarity scores, which increased even further when WordNet was applied. The imbalance between false positive and false negative values can also be attributed to class imbalance in the dataset.

5 Conclusion

In conclusion, this study evaluated the performance of three text representation models, IndoBERT, ELMo, and FastText, along with their combinations with WordNet, in the context of a CGAA exam simulation system. The combination of IndoBERT and WordNet model achieved the highest accuracy (0.6025), outperforming the standalone IndoBERT (0.565), FastText (0.245), and ELMo (0.2225). Although WordNet integration generally improved model accuracy, the overall accuracy of all configurations remained below 0.7. For performance evaluation, load testing was conducted using Apache JMeter to assess the response time of the API under 10 and 100 virtual users. The IndoBERT-only implementation showed an average response time of 178.2 ms for 10 users and 8806 ms for 100 users. Meanwhile, the combination of IndoBERT with WordNet recorded higher response times, 418.8 ms for 10 users and 15064 ms for 100 users, due to additional processing required for text expansion. Limitations include modest accuracy levels and the lack of an Indonesian-specific ELMo model. For future work, it is recommended to fine-tune deep learning-based models on domain-specific datasets and explore hybrid combinations of multiple embedding models to improve text representation accuracy and system performance.

Acknowledgments

I would like to express my sincere gratitude to Universitas Gadjah Mada, particularly the Department of Electronic Engineering and Informatics, Vocational School, for the support and resources provided throughout the duration of this research. The facilities, guidance, and academic environment offered by the institution played a crucial role in the successful completion of this study.

Funding Information
The authors declare no funding was received for this study.

Conflict of Interest Statement
The authors declare no conflicts of interest.

Ethical Approval
This study did not involve human or animal subjects.

Data Availability
The dataset used is available at https://data.mendeley.com/datasets/6gp8m72s9p/1

References

[1] T. A. Roshinta and F. Rahutomo, "Analisis Aspek-Aspek Ujian Esai Daring Berbahasa Indonesia," in Seminar Nasional Terapan Riset Inovatif, Oct. 2016, pp. 645–654. doi: 10.17632/6gp8m72s9p.1.

[2] Prof. S. A. Nalawade, R. R. Shukla, P. Gurjar, H. Kumar, and H. Chavan, "Smart Essay Grading," International Journal for Research in Applied Science and Engineering Technology, vol. 11, no. 1, pp. 1094–1098, Jan. 2023, doi: 10.22214/ijraset.2023.48750.

[3] D. Ramesh and S. K. Sanampudi, "An Automated Essay Scoring Systems: A Systematic Literature Review," Artificial Intelligence Review, vol. 55, no. 3, pp. 2495–2527, Mar. 2022, doi: 10.1007/s10462-021-10068-2.

[4] T. Wahyuningsih, "Text Mining an Automatic Short Answer Grading (ASAG), Comparison of Three Methods of Cosine Similarity, Jaccard Similarity and Dice's Coefficient," Journal of Applied Data Sciences, vol. 2, no. 2, pp. 45–54, 2021.

[5] M. Chen and Y. Dong, "Design of Exercise Grading System Based on Text Similarity Computing," Mobile Information Systems, vol. 2022, pp. 1–7, Jul. 2022, doi: 10.1155/2022/4634903.

[6] R. Patil, S. Boit, V. Gudivada, and J. Nandigam, "A Survey of Text Representation and Embedding Techniques in NLP," IEEE Access, vol. 11, 2023, doi: 10.1109/ACCESS.2023.3266377.

[7] M. Putnikovic and J. Jovanovic, "Embeddings for Automatic Short Answer Grading: A Scoping Review," IEEE Transactions on Learning Technologies, vol. 16, no. 2, pp. 219–231, Apr. 2023, doi: 10.1109/TLT.2023.3253071.

[8] Ikatan Akuntan Indonesia, "Informasi Umum US-CGAA," web.iaiglobal.or.id. Accessed: Feb. 18, 2025. [Online]. Available: https://web.iaiglobal.or.id/Sertifikasi-IAI/Informasi%20Umum%20US-CGAA#gsc.tab=0

[9] 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, 2022, doi: 10.11591/eei.v11i3.3531.

[10] L. Zhang, Y. Huang, X. Yang, S. Yu, and F. Zhuang, "An Automatic Short-answer Grading Model for Semi-open-ended Questions," Interactive Learning Environments, vol. 30, no. 1, pp. 177–190, Jan. 2022, doi: 10.1080/10494820.2019.1648300.

[11] M. R. R. Susanto, Husni Thamrin, and Naufal Azmi Verdikha, "Performance of Text Similarity Algorithms for Essay Answer Scoring in Online Examinations," Jurnal Teknik Informatika (Jutif), vol. 4, no. 6, pp. 1515–1521, Dec. 2023, doi: 10.52436/1.jutif.2023.4.6.1025.

[12] M. Kaya and I. Cicekli, "A Hybrid Approach for Automated Short Answer Grading," IEEE Access, vol. 12, pp. 96332–96341, 2024, doi: 10.1109/ACCESS.2024.3420890.

[13] M. C. Wijaya, "Automatic Short Answer Grading System in Indonesian Language Using BERT Machine Learning," Revue d'Intelligence Artificielle, vol. 35, no. 6, pp. 503–509, Dec. 2021, doi: 10.18280/ria.350609.

[14] H. M. Balaha and M. M. Saafan, "Automatic Exam Correction Framework (AECF) for the MCQs, Essays, and Equations Matching," IEEE Access, vol. 9, pp. 32368–32389, 2021, doi: 10.1109/ACCESS.2021.3060940.

[15] M. Hendre, P. Mukherjee, R. Preet, and M. Godse, "Efficacy of Deep Neural Embeddings-Based Semantic Similarity in Automatic Essay Evaluation," International Journal of Cognitive Informatics and Natural Intelligence, vol. 17, no. 1, pp. 1–14, May 2023, doi: 10.4018/IJCINI.323190.

[16] M. Alobed, A. M. M. Altrad, and Z. B. A. Bakar, "A Comparative Analysis of Euclidean, Jaccard and Cosine Similarity Measure and Arabic Wordnet for Automated Arabic Essay Scoring," in 2021 Fifth International Conference on Information Retrieval and Knowledge Management (CAMP), IEEE, Jun. 2021, pp. 70–74. doi: 10.1109/CAMP51653.2021.9498119.

[17] F. Lan, "Research on Text Similarity Measurement Hybrid Algorithm with Term Semantic Information and TF-IDF Method," Advances in Multimedia, vol. 2022, 2022, doi: 10.1155/2022/7923262.

[18] M. Barbouch, S. Verberne, and T. Verhoef, "WN-BERT: Integrating WordNet and BERT for Lexical Semantics in Natural Language Understanding," 2021. [Online]. Available: https://github.com/mbarbouch/WN-BERT.

[19] M. Kurniawan, K. Kusrini, and M. R. Arief, "Part of Speech Tagging Pada Teks Bahasa Indonesia dengan BiLSTM, CNN, CRF, dan ELMo," Jurnal Eksplora Informatika, vol. 11, no. 1, pp. 29–37, Jan. 2022, doi: 10.30864/eksplora.v11i1.506.