*Published online on the website: **https://journal.unilak.ac.id/index.php/dz***

Jurnal Teknologi Informasi & Komunikasi

Volume 16 Issue 1 Year 2025 | Page 1-11 | e-ISSN: 2477-3255 | ISSN: 2086-4884
Received: 24-03-2025 | Revised: 12-04-2025 | Accepted: 29-04-2025

Towards an Automated Essay Evaluation System NLP Based Text Embeddings and Similarity Metrics

Regita Putri Permata[1], Rendika Nurhartanto Suharto[2], Luh Candra Ayu Julianty[3]
[1,2,3]Data Science, Telkom University, Campus Surabaya, Ketintang 156 Surabaya
*Corespondence: regitapermata@telkomuniversity.ac.id

Abstract: *This study aims to develop an automatic essay answer assessment system based on Natural Language Processing (NLP) to reduce the time and effort required for evaluation. The system uses Cosine Similarity and Manhattan Distance as evaluation metrics and implements two text embedding methods—Term Frequency-Inverse Document Frequency (TF-IDF) and Bag of Words (BoW)—to represent the user's answer text. The methodology begins with text processing and pre-processing, followed by embedding and similarity calculation between the user's answer and the reference text to generate an evaluation score categorized into three levels: good, sufficient, and poor. Based on Cohen's Kappa analysis, the kappa value for Cosine Similarity reaches 0.78, indicating high agreement between the Cosine TF-IDF and Cosine BoW methods. In contrast, Manhattan Distance yields a kappa value of -0.05, indicating a discrepancy between the two Manhattan-based methods. The evaluation results suggest that Cosine Similarity is more suitable, whereas Manhattan Distance is not relevant for this task. At the modeling stage, the best classification models are Decision Tree and Random Forest, each achieving an accuracy of 96.67%. Although Random Forest demonstrates a higher AUC than Decision Tree, it requires a longer training time. Overall, the system is considered effective for assessing essay answers with both purpose and consistency, offering potential applications in the field of education.*

Keywords: *Automatic Essay Scoring, Cosine Similarity, Educational Assessment , Manhattan Distance, Natural Language Processing*

1. Introduction

Essay-based assessments are widely utilized in education to evaluate students' understanding of the material taught. Traditionally, essay grading is conducted manually by instructors or evaluators who assess responses based on various linguistic and conceptual aspects. However, manual grading poses several challenges, particularly in terms of subjectivity and inconsistency, as scoring may vary among evaluators. Additionally, as the volume of student essays increases, the grading process becomes time-consuming, labor-intensive, and prone to evaluator fatigue, potentially affecting the accuracy and fairness of assessments [1], [2].

To overcome these challenges, Automatic Essay Scoring (AES) systems have been developed to assist in evaluating text-based responses, such as student essays or job application assessments, in a faster, more objective, and consistent manner [3], [4]. With the increasing number of students and the rapid digitalization of education, AES technology has emerged as a promising solution for assessing students' critical thinking and conceptual understanding in an efficient and scalable way [5].

Despite its advantages, developing an automated essay evaluation system is a complex task. Natural Language Processing (NLP) techniques are required to analyze and understand the meaning of the text provided by users [3]. One of the widely used methods in text similarity evaluation is Cosine Similarity, which measures the similarity between two texts by computing the cosine angle between their vector representations. This technique helps determine how closely a student's answer aligns with an ideal reference response.

Several previous studies have explored different approaches to automated essay scoring (AES). One study applied the Winnowing algorithm, measuring text similarity through fingerprint generation and the Jaccard coefficient, which was implemented in an e-learning platform for essay grading [6]. Another study used TF-IDF weighting and the Vector Space Model to assess text similarity, supporting educators in grading student essays more effectively [7]. In addition, deep learning-based models have been introduced for improving essay evaluation in the Indonesian language, offering newer approaches to AES systems[8]. These studies demonstrate that a variety of methods—ranging from text similarity algorithms to deep learning techniques—have been investigated to enhance the efficiency and accuracy of AES systems.

However, many of these approaches still face limitations in accuracy, scalability, and reliability when evaluating the deeper semantic meaning of essays. Moreover, prior studies tend to focus on a single similarity metric or basic text representations, and few explicitly consider the consistency between automated scoring and human evaluations. In contrast, our system integrates TF-IDF and Bag of Words (BoW) embeddings with multiple evaluation metrics including Cosine Similarity, Manhattan Distance, and Cohen's Kappa to improve assessment precision and reliability. The inclusion of Cohen's Kappa enables measurement of inter-rater agreement, an aspect that has been largely overlooked in earlier research, thereby enhancing alignment between automated evaluations and human grading standards. By combining diverse embedding techniques and evaluation metrics, the proposed system offers a more robust, scalable, and objective solution for automated essay grading, with the potential to surpass previous research in both accuracy and consistency.

Based on these considerations, this study aims to address key research questions concerning the development of an automatic essay scoring system. First, it investigates how such a system can be designed to evaluate essays efficiently, consistently, and objectively using two distinct embedding methods: TF-IDF and Bag of Words (BoW). Second, it examines the application of Natural Language Processing (NLP) techniques to analyze and interpret essay responses, employing Cosine Similarity and Manhattan Distance as evaluation metrics. Finally, this research explores the integration of these similarity measures with supervised learning approaches to enhance the efficiency and consistency of essay evaluation. While the novelty of this approach is clear, there remains a lack of explicit comparative analysis with previous AES studies, which will be an important direction for future work to strengthen the validation of this system's performance.

2. Method

This study develops an Automatic Essay Scoring (AES) system using Natural Language Processing (NLP) with TF-IDF and Bag of Words (BoW) embeddings, evaluated through Cosine Similarity, Manhattan Distance, and Cohen's Kappa. The methodology consists of several key stages, as outlined below.

2.1 Data Collection and Preprocessing

A dataset of student essay responses is collected, each paired with a reference answer (gold standard) and manually assigned scores by human evaluators. The text undergoes preprocessing to improve data quality, including tokenization, stopword removal, lemmatization, lowercasing, and punctuation removal to eliminate inconsistencies.

Table 1. Essay Evaluation Dataset

Column	Data Type	Description
Question	String	Questions asked to users. For example, "Explain the impacts of global warming."
Reference Answer	String	Reference answers that have been assessed, used as a reference to evaluate user answers.

Column	Data Type	Description
User Answer	String	Answers written by users to given questions.

The following are the results of text preprocessing presented in Figure 1. There are 100 reference answers and one sample user answer.

	Reference Answer
0	Internet memungkinkan akses informasi dengan c...
1	Memulai bisnis online memerlukan perencanaan p...
2	Kesehatan mental penting agar kita dapat menja...
3	Perubahan iklim menyebabkan bencana alam, keru...
4	Teknologi mengubah proses kerja menjadi lebih ...
..	...
95	Penelitian ilmiah memperdalam pengetahuan dan ...
96	Menghormati orang tua menunjukkan kasih sayang...
97	Energi terbarukan ramah lingkungan dan membant...
98	Rasa percaya diri ditingkatkan dengan mengharg...
99	Teknologi mempermudah akses informasi dan memb...

(a)

	reference_answer_processed
0	internet ak informasi cepat komunikasi bata
1	bisnis online perencanaan produk platform penj...
2	kesehatan mental menjalani hidup produktif
3	perubahan iklim menyebabkan bencana alam kerus...
4	teknologi mengubah prose kerja efisien pekerja...
..	...
95	penelitian ilmiah memperdalam pengetahuan solu...
96	menghormati orang tua kasih sayang menghargai ...
97	energi terbarukan ramah lingkungan membantu me...
98	percaya ditingkatkan menghargai berlatih fokus...
99	teknologi mempermudah ak informasi pembelajara...

(b)

Figure 1. Preprocessing Text (a) Before (b) After

2.2 Text Embedding Representation

The natural language processing method known as word embedding enables the representation of word vectors in a multidimensional space. This representation is made possible by models such as Word2Vec, which allow the system to calculate semantic similarities between words, even if their forms differ. Word embedding enhances the system's ability to understand essay text more effectively. This leads to improved semantic analysis, enabling the system to assess the relevance and quality of essays more accurately [9], [10].

The TF-IDF method converts text into numerical vectors to measure similarity. First, the reference and user answers undergo preprocessing, including cleaning, lowercasing, and removing stop words. Next, both sets of answers are combined and vectorized using TfidfVectorizer(), which assigns weights based on term frequency and document rarity. The resulting TF-IDF matrix is then split into reference and user answer vectors.

Figure 2 represents a 100×491 TF-IDF matrix, where each row corresponds to a processed answer (reference or user), and each column represents a unique term in the vocabulary. The values indicate TF-IDF scores, with many zeros due to sparse data. Nonzero values (e.g., 0.39335708 and 0.36538011) highlight important terms. This matrix is used to measure text similarity between reference and user answers.

```
[[0.         0.         0.39335708 ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]
 ...
 [0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.36538011 ... 0.         0.         0.        ]]
```

Figure 2. TF-IDF Matrices

```
[[0 0 1 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]]
```

Figure 3. BoW Matrices

Figure 3 represent of the 100×491 BoW (Bag of Words) matrix represents the frequency of each word in the processed answers, where each row corresponds to an answer (reference or user), and each column represents a unique word from the vocabulary. Unlike TF-IDF, BoW only captures word occurrences without weighting their importance [11], [12], [13]. The resulting matrix is sparse, with many zero values indicating the absence of words, while nonzero values represent

word counts. This matrix is useful for text similarity evaluation based on word presence rather than semantic relevance.

2.3 Similarity Evaluation Using NLP Metrics

To measure the similarity between a student's answer and the reference, Cosine Similarity and Manhattan Distance are used. Cosine Similarity computes the angle between two text vectors, indicating how closely a student's response aligns with the reference answer. Meanwhile, Manhattan Distance calculates the absolute differences between corresponding feature values in the text vectors. These metrics help determine textual similarity from different perspectives [14]. The similarity evaluation calculation also applies to BoW feature extraction.

```
Cosine Similarity (TF-IDF):
 [[0.40913723 0.         0.         ... 0.         0.         0.14700554]
 [0.         0.80407461 0.         ... 0.         0.         0.        ]
 [0.         0.         0.86823912 ... 0.         0.         0.        ]
 ...
 [0.         0.         0.         ... 0.98051413 0.         0.        ]
 [0.         0.         0.         ... 0.         0.26578053 0.        ]
 [0.0969573  0.         0.         ... 0.         0.         0.59287672]]
```

(a)

```
Manhattan Distance (TF-IDF):
 [[3.02933299 5.23851634 5.01510216 ... 4.94335358 4.83329678 4.08622177]
 [5.61646504 1.13848576 5.58052507 ... 5.50877649 5.39871969 5.39908537]
 [4.82451219 5.0119864  0.97577206 ... 4.71682364 4.60676684 4.60713251]
 ...
 [5.29724115 5.48471536 5.26130118 ... 0.2455747  5.0794958  5.07986148]
 [5.063441   5.25091521 5.02750102 ... 4.95575245 3.44509199 4.84606132]
 [4.64833954 5.42795584 5.20454166 ... 5.13279308 5.02273628 2.03592884]]
```

(b)

Figure 4. Similarity Evaluation (a) Cosine Similarity TF-IDF (b) Manhattan Distance TF-IDF

2.4 Inter-Rater Agreement Measurement

To ensure consistency between automated scoring and human grading, Cohen's Kappa is applied. This statistical measure evaluates the level of agreement between human and system-generated scores, correcting for agreement occurring by chance. A higher Cohen's Kappa value indicates better reliability of the system in replicating human scoring patterns [15].

2.5 Supervised Learning for Classification

Using similarity scores from Cosine Similarity and Manhattan Distance, Decision Tree, Random Forest and several models are trained to classify responses into three categories: Good, Sufficient, and Poor. These models predict essay scores based on extracted features and are evaluated using accuracy, precision, recall, F1-score, and AUC to ensure their effectiveness.

2.6 System Evaluation and Comparative Analysis

The system's performance is compared against existing AES methods, including TF-IDF with Jaccard Similarity and deep learning-based approaches. The results demonstrate that incorporating TF-IDF, BoW, Cosine Similarity, and Cohen's Kappa improves accuracy and reliability, making this method a more robust and scalable solution for automated essay evaluation.

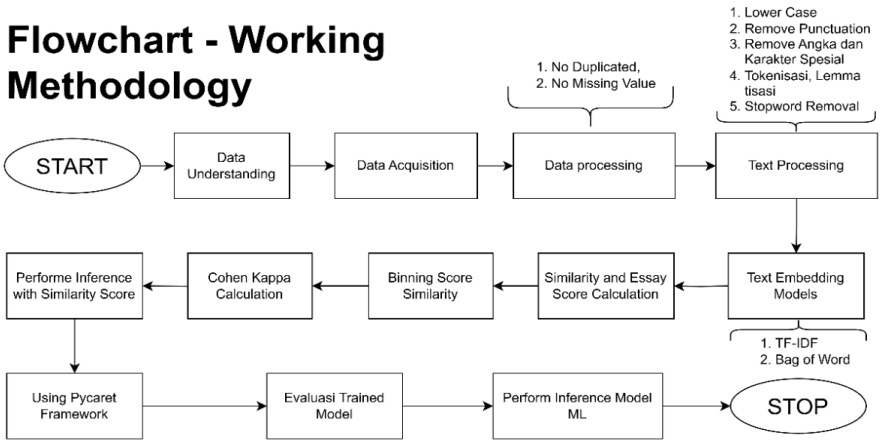


Figure 5. Flowchart – Working Methodology

3. Results and Discussion

3.1 Similarity and Essay Score Calculation

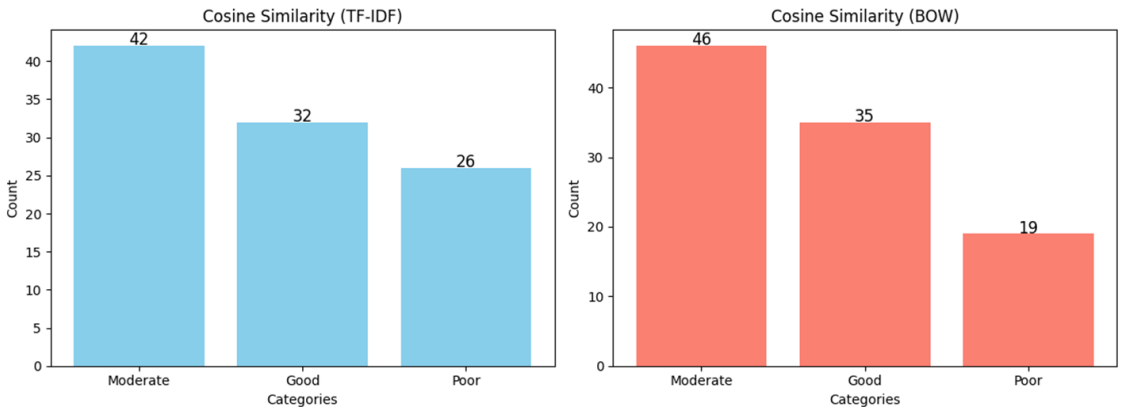
Figure 4 (2.3) illustrates the process of performing similarity calculations between reference and user answers using two different metrics: Cosine Similarity and Manhattan Distance, applied to vector representations generated by both TF-IDF and Bag of Words (BoW) methods. First, Cosine Similarity is used to measure the degree of similarity between reference and user answer vectors, where values closer to 1 indicate a higher degree of similarity. Next, Manhattan Distance is computed using the cityblock function, which calculates the absolute difference between vector points in a multidimensional space; lower values indicate greater similarity. The results from both metrics provide meaningful insights into assessing the relevance and quality of essay responses, contributing to a more objective and consistent automated evaluation system.

The binning functions categorize similarity and distance values into three qualitative labels: 'Good' (Good), 'Moderate' (Moderate), and 'Bad' (Poor). The bin_similarity function assigns a 'Good' label for similarity values ≥ 0.70, 'Moderate' for values between 0.5 and 0.69, and 'Bad' for values below 0.5. Meanwhile, the bin_distance function evaluates distance metrics, where values ≤ 3 are categorized as 'Good', values between 4 and 7 as 'Moderate', and values > 7 as 'Bad'. These classifications help interpret evaluation metrics by grouping numerical results into qualitative performance levels for better readability and analysis.

	Question	Reference Answer	User Answer	Similarity (Cosine TF-IDF)	Similarity (Cosine BOW)	Distance (Manhattan TF-IDF)	Distance (Manhattan BOW)	Cat_Similarity (Cosine TF-IDF)	Cat_Similarity (Cosine BOW)	Cat_Distance (Manhattan TF-IDF)	Cat_Distance (Manhattan BOW)
13	Bagaimana cara efektif mengatur keuangan pribadi?	Mengatur keuangan membutuhkan pencatatan penge...	Pengelolaan keuangan dapat dilakukan dengan me...	0.552437	0.571429	2.359069	6	Moderate	Moderate	Good	Moderate
46	Bagaimana cara meningkatkan daya tahan tubuh?	Daya tahan tubuh dapat ditingkatkan dengan mak...	Mengonsumsi makanan sehat, berolahraga, dan is...	0.423735	0.471405	3.314587	9	Poor	Poor	Moderate	Poor
19	Apa fungsi teknologi blockchain?	Blockchain adalah teknologi yang menyimpan dat...	Blockchain menyimpan data secara aman dan tran...	0.858820	0.857143	0.897793	2	Good	Good	Good	Good

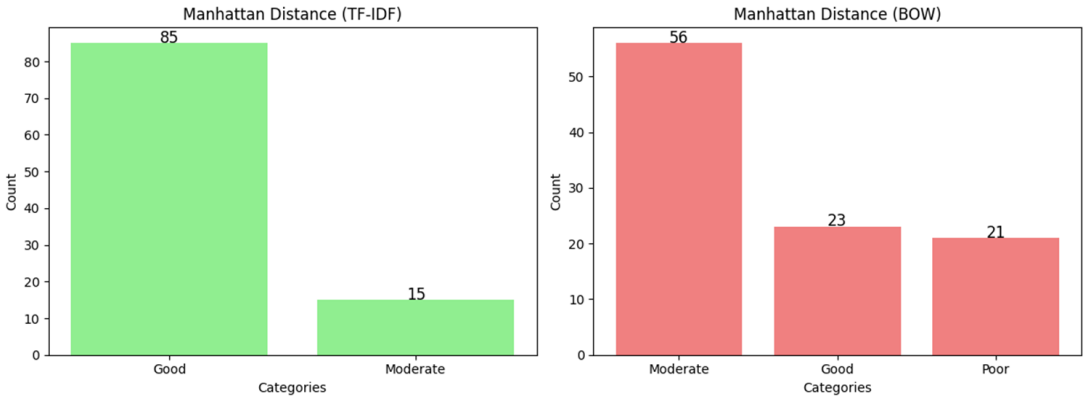
Figure 6. Similarity and Distance Values

The figure 6 presents a table where similarity and distance values between reference and user answers are categorized into qualitative labels (‘Good’, ‘Moderate’, ‘Bad’) using the binning functions, demonstrating how different vectorization methods (TF-IDF and BoW) impact similarity and distance scores, which in turn influence the classification of responses.



(a) (b)

Figure 7. Category Distribution based of Cosine Similarity (a) TF-IDF (b) BoW



(a) (b)

Figure 8. Category Distribution based of Manhattan Similarity (a) TF-IDF (b) BoW

Figure 7 and Figure 8 above is a bar chart that illustrates the number of labels generated by various methods. The Cosine Similarity (Figure 7) results indicate that both TF-IDF and BoW categorize most responses as "Moderate" or "Good," with TF-IDF showing a more balanced distribution across categories, while BoW has a higher concentration in the "Moderate" category. Meanwhile, Manhattan Distance (Figure 8) results reveal that TF-IDF overwhelmingly classifies responses as "Good," whereas BoW shows a more even distribution across all three categories, suggesting differences in how each method measures semantic similarity.

Next, Cohen's Kappa analysis will be carried out to assess the level of agreement between the assessments generated by these methods. By measuring the Kappa coefficient, we can determine the extent to which the results obtained from each method are consistent with each other in providing labels [16].

Table 2. Cohen's Kappa Value

Cohen's Kappa for Similarity	0.78308026
Cohen's Kappa for Distance	0.05482304

Table 2 shows the results of the analysis using Cohen's Kappa that the Kappa value for Similarity is 0.78, which indicates a high level of agreement between the two similarity evaluation methods (Cosine TF-IDF and Cosine BoW), with the interpretation that the assessments given by the two methods tend to be consistent and show objectivity in the evaluation. In contrast, the Kappa value for Distance is -0.05, which indicates no agreement between the distance assessments (Manhattan TF-IDF and Manhattan BoW), even indicating that the assessments of the two methods may contradict each other, so it is necessary to pay attention again in choosing the right metric for distance evaluation in this context. So cosine similarity will be used for the results below.

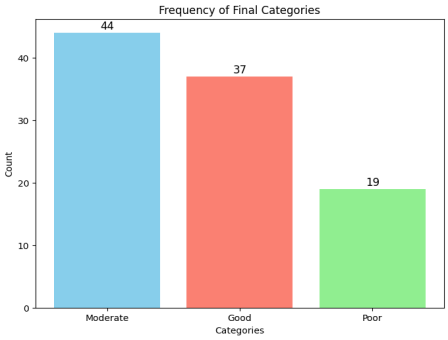


Figure 9. Category frecuency using Cosine Similarity

In Figure 9, the final category is determined using the majority voting method based on the categories generated from two methods, namely Cosine Similarity TF-IDF and Cosine Similarity BOW. The results are stored in the Final_Similarity_Category column, which is then used as the Final_Score in the results DataFrame for further analysis. By looking at the frequency of the final category, we can understand the distribution of the assessment results and identify general trends in the evaluation of essay answers carried out by the system.

3.2 Model Machine Learning Prediction

Table 3. Values for Machine Learning Input

Feature		**Output**
Similarity (Cosine TF-IDF)	*Similarity (Cosine BOW)*	*Final_Score*
0.600081	0.720577	Good
0.655111	0.666667	Fair
0.539374	0.597614	Fair
0.174867	0.204124	Bad
0.580774	0.555556	Fair

The table 3 is the value that will be entered into the machine learning algorithm that is expected to increase the effectiveness and intelligence of the model. Using PyCaret to prepare the classification process with previously prepared data, where the data division shows 75 samples used for the training set and 24 samples for the testing set. In addition, the stratified K-Fold method is used to ensure that the proportion of each class in the training and testing sets remains balanced. This is a multi-class classification problem with three labels, namely "Good," "Fair," and "Bad," each of which is mapped to a numeric value (Good: 0, Fair: 1, Bad: 2). This setup also includes the use of SMOTE to correct class imbalance, with the aim of improving model performance by providing more examples for underrepresented classes.

Table 4. Classification Algorithm Result

Rank	**Code Model**	**Model**	**Accuracy**	**AUC**	**Recall**	**Prec.**	**F1**	**Kappa**	**MCC**	**TT (Sec)**
1	**DT**	Decision Tree Classifier	0.967	0.967	0.967	0.958	0.958	0.941	0.950	0.043
2	**RF**	Random Forest Classifier	0.967	10.000	0.967	0.958	0.958	0.941	0.950	0.355
3	**ADA**	Ada Boost Classifier	0.967	0.000	0.967	0.958	0.958	0.941	0.950	0.149
4	**GBC**	Gradient Boosting Classifier	0.967	0.000	0.967	0.958	0.958	0.941	0.950	0.212
5	**ET**	Extra Trees Classifier	0.967	10.000	0.967	0.958	0.958	0.941	0.950	0.166
6	**KNN**	K Neighbors Classifier	0.950	0.976	0.950	0.968	0.950	0.920	0.931	0.055
7	**LDA**	Linear Discriminant Analysis	0.950	0.000	0.950	0.968	0.950	0.920	0.931	0.042

Rank	Code Model	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
8	**Xgboost**	Extreme Gradient Boosting	0.950	0.961	0.950	0.947	0.941	0.915	0.927	0.077
9	**Lightgbm**	Light Gradient Boosting Machine	0.950	0.990	0.950	0.938	0.943	0.916	0.920	0.235
10	**NB**	Naive Bayes	0.930	0.994	0.930	0.957	0.931	0.893	0.903	0.045
11	**QDA**	Quadratic Discriminant Analysis	0.917	0.000	0.917	0.953	0.917	0.876	0.891	0.080
12	**LR**	Logistic Regression	0.860	0.000	0.860	0.918	0.851	0.791	0.821	0.531
13	**SVM**	SVM - Linear Kernel	0.803	0.000	0.803	0.792	0.777	0.710	0.739	0.047
14	**Ridge**	Ridge Classifier	0.557	0.000	0.557	0.336	0.411	0.369	0.484	0.039
15	**Dummy**	Dummy Classifier	0.377	0.500	0.377	0.145	0.208	0.000	0.000	0.060

From Table 4 shows the best classification models in this result are Decision Tree Classifier and Random Forest Classifier, both have the highest accuracy of 96.67%. However, Random Forest shows a better AUC (10,000) than Decision Tree which has an AUC of 0.9667, although the stated AUC seems unrealistic and needs to be further examined. Both also show very good performance in other metrics, such as Recall, Precision, and F1, all of which are close to the optimal value. However, Random Forest has a longer training time (0.355 seconds) than Decision Tree (0.043 seconds), which can be an important factor in applications that require time efficiency. Overall, the best choice depends on the context of the application, but if we focus on better performance metrics and more informative AUC, Random Forest is the superior choice.

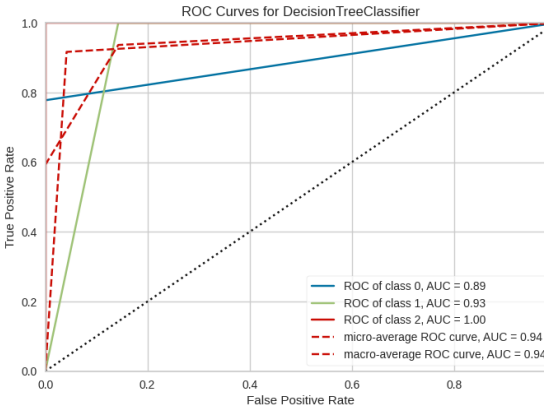


Figure 10. ROC Curves for DecisionTree Classifier

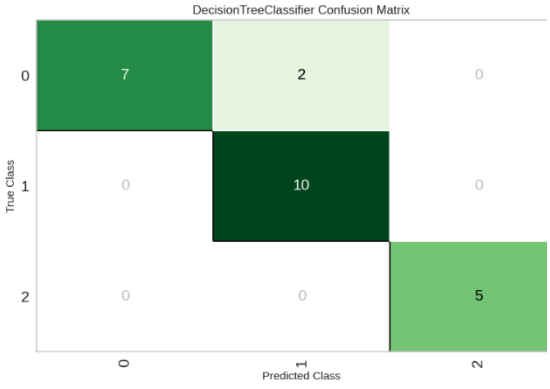


Figure 11. DecisionTree Classifier Confusion Matrix

Figures 10 and 11 present the evaluation results of the Decision Tree Classifier model. In Figure 9 (left), the ROC curve plot illustrates the model's performance in distinguishing between classes, using AUC (Area Under the Curve) as the primary metric. The AUC scores are 0.89 for class 0,

0.93 for class 1, and 1.00 for class 2, indicating that the model performs very well in classifying class 2. The overall macro-average and micro-average AUC values are both 0.94, reflecting a strong overall performance across all classes. Figure 10 (right) shows the corresponding confusion matrix, where the model correctly classifies 22 out of the total samples. However, some misclassifications occur—specifically, two samples from class 0 are incorrectly classified as class 1. Despite these errors, the Decision Tree Classifier demonstrates good accuracy and consistent classification performance across multiple categories.

5. Conclusions

This study successfully applied various text processing and embedding techniques to transform textual data into vector representations for further analysis. The preprocessing stage, which involved lemmatization, stop word removal, and punctuation elimination, effectively enhanced data quality for natural language processing tasks. The TF-IDF and Bag of Words (BoW) methods consistently generated text representations, yielding a unique feature count of 491, demonstrating structural consistency in the processed data.

Furthermore, Cosine Similarity and Manhattan Distance were employed to measure the alignment between reference and user responses. The results indicated that Cosine Similarity, particularly with TF-IDF and BoW, achieved a high level of consistency (Cohen's Kappa: 0.78), making it a reliable metric for text similarity assessment. In contrast, Manhattan Distance produced less consistent results (Kappa: -0.05), suggesting that Cosine Similarity is the more dependable approach in this study.

In the machine learning evaluation, Decision Tree and Random Forest emerged as the top-performing classification models. Random Forest achieved an accuracy of 96.67%, an AUC of 1.000 (requiring further verification), a recall of 96.67%, a precision of 95.83%, an F1-score of 95.79%, Cohen's Kappa of 94.14%, and an MCC of 95.00%. Similarly, the Decision Tree model exhibited an accuracy of 96.67% and an AUC of 0.9667, with comparable recall, precision, F1-score, Cohen's Kappa, and MCC values. Despite its faster training time (0.043 seconds vs. 0.355 seconds for Random Forest), Random Forest was selected as the superior model due to its overall stronger performance metrics.

References

[1] A. Kayan, A. Sanjaya, and U. Mahdiyah, "Koreksi Otomatis Ujian Esai Menerapkan Algoritma Winnowing Dan Metode Cosine Similarity," 2024.

[2] R. Fitri and A. N. Asyikin, "Aplikasi Penilaian Ujian Essay Otomatis Menggunakan Metode Cosine Similarity," *POROS Tek.*, vol. 7, no. 2, pp. 88–94, 2015, doi: 10.31961/porosteknik.v7i2.218.

[3] D. O. Sihombing, "Implementasi Natural Language Processing (NLP) dan Algoritma Cosine Similarity dalam Penilaian Ujian Esai Otomatis," *J. Sist. Komput. Dan Inform. JSON*, vol. 4, no. 2, p. 396, Dec. 2022, doi: 10.30865/json.v4i2.5374.

[4] K. Sun and R. Wang, "Automatic Essay Multi-dimensional Scoring with Fine-tuning and Multiple Regression," Jun. 03, 2024, *arXiv*: arXiv:2406.01198. doi: 10.48550/arXiv.2406.01198.

[5] K. Doi, K. Sudoh, and S. Nakamura, "Automated Essay Scoring Using Grammatical Variety and Errors with Multi-Task Learning and Item Response Theory," in *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, E. Kochmar, M. Bexte, J. Burstein, A. Horbach, R. Laarmann-Quante, A. Tack, V. Yaneva, and Z. Yuan, Eds., Mexico City, Mexico: Association for Computational Linguistics, Jun. 2024, pp. 316–329. Accessed: Mar. 24, 2025. [Online]. Available: https://aclanthology.org/2024.bea-1.26/

[6] S. Astutik, A. D. Cahyani, and M. K. Sophan, "Sistem Penilaian Esai Otomatis Pada E-Learning Dengan Algoritma Winnowing," *J. Inform.*, vol. 12, no. 2, pp. 47–52, Jan. 2014, doi: 10.9744/informatika.12.2.47-52.

[7] M. Mi'andri, A. C. Siregar, and P. Y. Utami, "Sistem Penilaian Ujian Otomatis Untuk Soal Esai Menggunakan Metode Vector Space ModeL," *JUTECH J. Educ. Technol.*, vol. 2, no. 2, pp. 1–15, Jan. 2022, doi: 10.31932/jutech.v2i2.1273.

[8] I. Huda, "Penerapan Deep Learning pada Kasus Sistem Penilaian Esai Otomatis Bahasa Indonesia," 2022, Accessed: Mar. 24, 2025. [Online]. Available:

https://digilib.uns.ac.id/dokumen/89030/Penerapan-Deep-Learning-pada-Kasus-Sistem-Penilaian-Esai-Otomatis-Bahasa-Indonesia

[9] K. Poonpon, P. Manorom, and W. Chansanam, "Exploring effective methods for automated essay scoring of non-native speakers," *Contemp. Educ. Technol.*, vol. 15, no. 4, p. ep475, Oct. 2023, doi: 10.30935/cedtech/13740.

[10] M. Meccawy, A. A. Bayazed, B. Al-Abdullah, and H. Algamdi, "Automatic Essay Scoring for Arabic Short Answer Questions using Text Mining Techniques," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 6, 2023, doi: 10.14569/IJACSA.2023.0140682.

[11] D. Sugiarto, E. Utami, and A. Yaqin, "Perbandingan Kinerja Model TF-IDF dan BOW untuk Klasifikasi Opini Publik Tentang Kebijakan BLT Minyak Goreng," *J. Tek. Ind.*, vol. 12, no. 3, Art. no. 3, Dec. 2022, doi: 10.25105/jti.v12i3.15669.

[12] I. F. Effendi, D. A. Utami, R. A. Rahmawati, R. Prasetyowibowo, and P. Isbandono, "Twitter Data Sentiment Analysis on the Economic Sector: Public Response to Government Policies During the COVID-19 Pandemic in Indonesia," presented at the International Joint Conference on Arts and Humanities 2023 (IJCAH 2023), Atlantis Press, Dec. 2023, pp. 472–491. doi: 10.2991/978-2-38476-152-4_45.

[13] K. T. Putra, M. A. Hariyadi, and C. Crysdian, "Perbandingan Feature Extraction Tf-Idf Dan Bow Untuk Analisis Sentimen Berbasis Svm".

[14] F. Li, X. Xi, Z. Cui, D. Li, and W. Zeng, "Automatic Essay Scoring Method Based on Multi-Scale Features," *Appl. Sci.*, vol. 13, no. 11, Art. no. 11, Jan. 2023, doi: 10.3390/app13116775.

[15] S. M. Vieira, U. Kaymak, and J. M. C. Sousa, "Cohen's kappa coefficient as a performance measure for feature selection," in *International Conference on Fuzzy Systems*, Jul. 2010, pp. 1–8. doi: 10.1109/FUZZY.2010.5584447.

[16] G. Rau and Y.-S. Shih, "Evaluation of Cohen's kappa and other measures of inter-rater agreement for genre analysis and other nominal data," *J. Engl. Acad. Purp.*, vol. 53, p. 101026, Sep. 2021, doi: 10.1016/j.jeap.2021.101026.