computers

Article

Enhancing GAN-LCS Performance Using an Abbreviations Checker in Automatic Short Answer Scoring

Ar-Razy Muhammad [1,2], Adhistya Erna Permanasari [2] and Indriana Hidayah [2,*]

[1] Department of Informatics Engineering, Politeknik Negeri Ketapang, Ketapang 78851, Indonesia; ar-razy.muhammad@politap.ac.id
[2] Department of Electrical and Information Engineering, Faculty of Engineering, Universitas Gadjah Mada, Yogyakarta 55281, Indonesia; adhistya@ugm.ac.id
* Correspondence: indriana.h@ugm.ac.id; Tel.: +62-822-2550-6006

Abstract: Automatic short answer scoring methods have been developed with various algorithms over the decades. In the Indonesian language, the string-based similarity is more commonly used. This method is difficult to accurately measure the similarity of two sentences with significantly different word lengths. This problem has been handled by the Geometric Average Normalized-Longest Common Subsequence (GAN-LCS) method by eliminating non-contributive words utilizing the Longest Common Subsequence method. However, students' answers may vary not only in character length but also in the words they choose. For instance, some students tend only to write the abbreviations or acronyms of the phrase instead of writing meaningful words. As a result, it will reduce the intersection character between the reference answer and the student answer. Moreover, it can change the sentence structure even though it has the same meaning by definition. Therefore, this study aims to improve GAN-LCS method performance by incorporating the abbreviation checker to handle the abbreviations or acronyms found in the reference answer or student answer. The dataset used in this study consisted of 10 questions with 1 reference answer for each question and 585 student answers. The experimental results show an improvement in GAN-LCS performance that could run 34.43% faster. Meanwhile, the Root Mean Square Error (RSME) value became lower by 7.65% and the correlation value was increased by 8%. Looking forward, future studies may continue to investigate a method for automatically generate the abbreviations dictionary.

Keywords: automatic short answer scoring; sentence similarity scoring; abbreviations checker; query expansion

check for updates

Citation: Muhammad, A.-R.; Permanasari, A.E.; Hidayah, I. Enhancing GAN-LCS Performance Using an Abbreviations Checker in Automatic Short Answer Scoring. *Computers* **2022**, *11*, 108. https://doi.org/10.3390/computers11070108

Academic Editors: Samir Garbaya, George A. Papadopoulos and Paolo Bellavista

Received: 16 March 2022
Accepted: 29 June 2022
Published: 1 July 2022

1. Introduction

The COVID-19 pandemic has had a significant impact on a number of global issues that directly affect human life, including health, tourism, and the economy. After the health sector, the education sector has been the most impacted by COVID-19 [1]. During the pandemic, distance learning became mainstream [2] and the educational system switched to online classes [3]. This transformation also significantly increases MOOC enrollment and encourages massive MOOC implementation [4]. Not only that, assessments have also been conducted through computer-based systems. The assessments can take the form of multiple-choice, short answer, and essay that are scored either manually or automatically calculated by the computer systems [5].

In relation to the automatic scoring technique, automatic essay scoring has evolved with various algorithms in the last few decades. By using the automatic system, students' answers can be efficiently and fairly evaluated by the algorithm based on the teacher's reference answer [5]. There are several advantages to using the automatic approach, including the ability to assess student answers faster [6] and being more objective and consistent [7].

Automatic essay scoring can be categorized into two categories: Automated Short Answer Scoring (ASAS) and Automated Essay Scoring (AES). What makes them different is the length of the answer. Automatic Short Answer Scoring can only be used when the length of the answer is not less than one phrase, and can possibly assess one paragraph [5] that contains four sentences [8]. However, Burrow states [5] that, before 2015, there were 35 methods for ASAS, but none of them perform as well as humans do. This situation encourages many researchers to work harder so that the system can show more accurate results. On the other side, Automatic Essay Scoring can be implemented to assess longer series of sentences. In addition, some previous studies discussing the Indonesian language focused more on the AES rather than ASAS.

Furthermore, certain methods that frequently used were String-Based Similarity, namely, Cosine Similarity [9–12], Latent Semantic Analysis (LSA) [13–17], Winnowing Algorithm [18], and Semantic Text Similarity (STS) [19].

Those methods, also, can hardly handle the short answer case. String-Based Similarity requires keywords or terms of the student answer to be identical to the reference answer. In short answer cases, the keywords or terms used may only appear once or even not appear at all [12]. Furthermore, String-Based Similarity methods have great difficulty to accurately measure the similarity of two sentences that consist of different words or character lengths. Pribadi [20] proposed a new method named Geometric Average Normalized-Longest Common Subsequence (GAN-LCS). It is a type of Character-Based Similarity approach that can be used to handle the variety of student answers, which still have the same meaning but different character lengths. It utilizes the variations of reference answers with automatic generation using Maximum Marginal Relevance (MMR).

Notwithstanding, the students' answers could vary the character length of the word they chose; some students would often write only the abbreviations or the acronyms. Consequently, GAN-LCS accuracy will be decreased. As it is known, GAN-LCS uses the intersection between the sequence of characters on references and student answers. When the intersection character between references and student answer is reduced, the accuracy of GAN-LCS is also decreased. To overcome this problem, this research proposed an abbreviations checker method to handle the abbreviations and acronyms that might be found in the reference answer or student answers.

2. Materials and Methods

2.1. Research Method

Figure 1 illustrates the research flow of our study. At first, data collection was collected from the questions and answers in a quiz on the Information System Analysis and Design course at the Department of Informatics Engineering, Politeknik Negeri Ketapang.

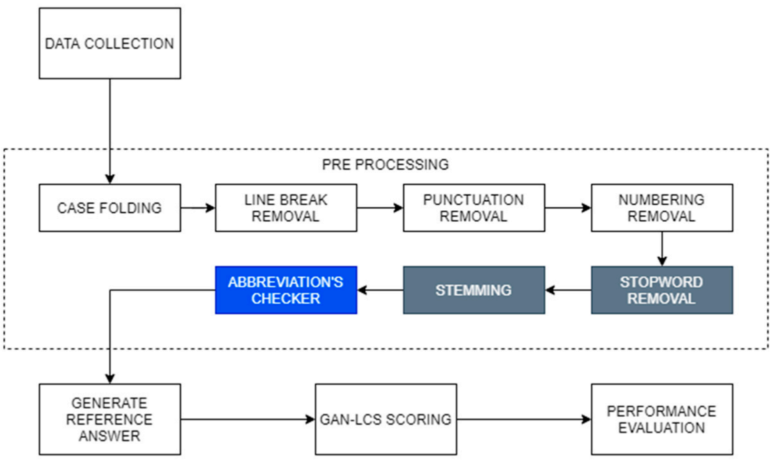

Figure 1. Research Flow.

Secondly, we applied a pre-processing method, which consisted of seven steps: case folding, line break removal, punctuation removal, numbering removal, stop-word removal, stemming, and the last pre-processing step was applying the proposed method called the abbreviations checker marked by the blue square in Figure 1. In the pre-processing step, we added a new approach different from the original one by using stop-word removal and stemming, marked in the gray square on Figure 1, in which these steps were not applied to the original method proposed by Pribadi [20].

The third and fourth steps were adopted from Pribadi's proposed method [20] by using MMR to generate variant reference answers and using the GAN-LCS method to calculate the final score. The last step was performance evaluation to measure the similarity between the score generated by the system and the score manually calculated by the teacher.

2.2. Dataset

The dataset consists of 10 questions with 1 teacher response to each question as a reference answer and 587 student answers in Indonesian that was scored by the teacher manually in the range from 0 to 10. Each question was answered by between 57 and 59 students. Table 1 shows an example of a dataset fragment consisting of questions, the teacher answer marked as the reference answer, student answers, and the score for each answer.

Table 1. Example of the dataset fragment.

Respondent	Answer	Score
* Teacher	DFD, Kamus Data, ERD	10
Student 1	1. DFD 2. Kamus Data 3. ERD	10
Student 2	1. DFD 2. Kamus Data 3. Entity Relationship Diagram 4. State Transition Diagram 5. Structured Chart 6. Diagram SADT	10
Student 3	1. DFD (Data Flow Diagram) 2. Kamus Data 3. Entity Relationship Diagram (ERD)	10
...	...	...
Student 59	DFD (Data Flow Diagram–Diagram Arus Data, DAD)Kamus DataEntity Relationship Diagram (ERD)	10

* Reference Answer.

2.3. Text Pre-Processing

The text pre-processing step is conducted to clean the answer from unwanted characters that can distract the calculation process. In this study, we perform several stages of the pre-processing method, consisting of:

1. Case folding is performed to convert all the characters into lowercase [21];
2. Line break removal is performed to convert a multi-line answer into a long single-line answer;
3. Punctuation removal is performed to remove all meaningless characters from the string;
4. Numbering removal is performed to convert a numbered list text into a long flattened string;

5. Stopword removal is performed to remove words included in the stopword list. The stopword list consists of the words that have a high chance of appearing in the string but are meaningless;
6. The stopword list used in this study comes from the Stopwords Indonesian (ID) repository by Stopwords ISO [22], consisting of 758 words based on Tala's research [23];
7. Stemming is a method to find the root word using several techniques [24]. The stemming library used in this study comes from the Sastrawi Repository [25]. This library is constructed based on the Nazief-Adriani Algorithm [26] as well as the algorithm based on Asian's [27], Arifin's [28], and Tahitoe's [29] research.

2.4. Abbreviations Dictionary

The abbreviations dictionary is a set of word lists consisting of abbreviations or acronyms and their definitions. Even though there is a ready-to-use third-party abbreviations dictionary, we were unable to use it in this study due to the possibility that an abbreviation has a different meaning in other domains.

Due to this, the manual approach is the best choice for generating the abbreviations dictionary. The teacher must construct the dictionary manually for each exam session by creating a list of terms and the definition that appears in the reference answer. Moreover, the term and definition in the dictionary must be written in lowercase because the scoring method we used in this study is a character-based approach; the capitalized character and the lowercase one will be recognized as different characters by the scoring system.

In this study, we used an abbreviations dictionary consisting of fifteen words based on our dataset. The dictionary was later used in the abbreviation checking process and compared to the teacher's answer and the student's answers.

Table 2 shows an example of an abbreviations dictionary that consists of the terms and the definition.

Table 2. Example of the dataset fragment.

Term	Definition
dad	diagram arus data
dfd	data flow diagram
erd	entity relationship diagram
...	...
sadt	structured analysis and design technique

2.5. Abbreviations Checker

The abbreviations checker is a method that is proposed to minimize the character length difference and increase the character intersection between two sentences without changing the meaning of the sentence by expanding the query that is used to get the more relevant result [30]. Figure 2 shows the process of the abbreviations checker and is explained as follows:

1. Import the abbreviations dictionary. This dictionary should be attached to the system;
2. Find abbreviations and acronyms in the reference answer. This step results in two outputs, first is the list of terms that are found in the reference answers. The second is the new abbreviations dictionary based on the abbreviations or acronyms that are used in the reference answer;
3. Extract the definition of the terms;
4. Append the definition to the next of the terms. If the definition has been written in the sentence and the term is not found in the sentence, the term will be inserted before the definition;
5. New final string is constructed;
6. The same process is performed for all student answers but with the new abbreviations dictionary that was created on process 2.

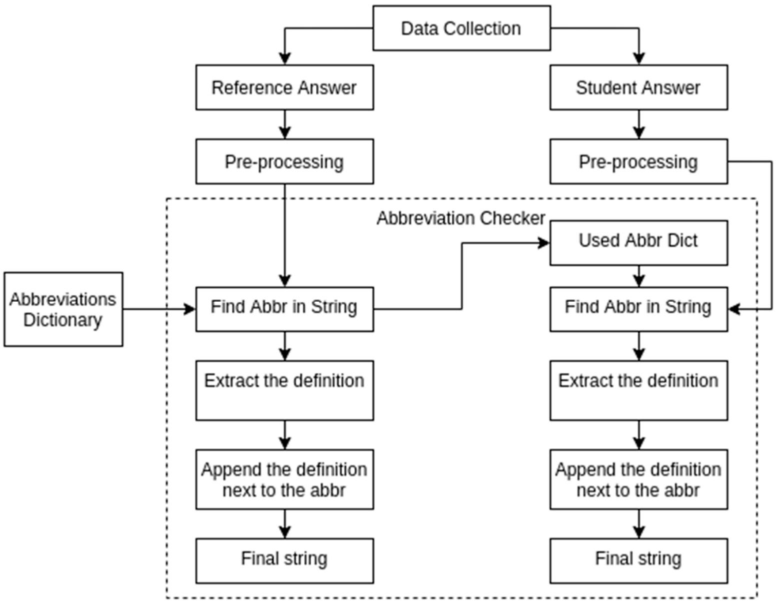

Figure 2. Abbreviations checker process.

2.6. Generate Reference Answer Using Maximum Marginal Relevance (MMR)

The ASAS scoring method requires a variety of reference answers to gain the best accuracy while dealing with the diversity of student answers [31,32]. The heterogeneity of student answers occurred because the students had their own way of constructing the sentences.

The variation of reference answers for much research has been conducted using various methods, including manual and automatic approaches. The manual approach that was used in [31,33] required the teacher to make more than one reference answer to cover the possibility of student answers using a different structure.

The automatic approach demonstrated by Mohler [34] automatically generates reference answers from the student answers. This method uses the Rocchio method to find student answers that are the most similar to the reference answer. However, the Rocchio method has a drawback because it requires a time-consuming training process.

Pribadi, on the other hand, proposed a new method for automatically generating alternative reference answers using Maximum Marginal Relevance (MMR). The MMR method is one of a text extraction document technique that can summarize one or many documents by rank and compares the similarities between the documents [35].

Equation (1) formulated the MMR method, with Q representing the reference answer, D_i representing the student answers, and D_J representing the student answer with the highest MMR score from the previous iteration. λ is a constant that can be used to adjust the relevance or diversity of sentences, with a value ranging from 0 to 1. The λ value that is closest to 1 means that the two sentences are more similar, and vice versa. In his paper, Pribadi uses a value of 0.85 based on [35] that is also implemented in this study. The result is a collection of different reference answers generated from available student answers that are the most similar to the teacher's answer.

In this study, we use three iterations and provide three variant reference answers. Since each question has its own reference answer from the teacher, the output of this process is four reference answers, including the original reference answer. *Sim*, in Equation (1), is the method to measure the similarity between sentences, such as D_i to Q or D_i to D_J, utilizing the Cosine Coefficient (*CC*) equation, as shown in (2). *CC* is a technique to measure the relevance between two sentences, with R as the first sentence and S as the second sentence.

$$MMR = \underset{D_i \in R \setminus S}{\operatorname{argmax}} \left[\lambda (Sim_1(D_i,\ Q)) - (1 - \lambda) \max_{D_j \in S} Sim_2(D_i, D_j) \right] \tag{1}$$

$$CC = \frac{R \cap S}{\sqrt{R}\ .\ \sqrt{S}} \tag{2}$$

2.7. Geometric Average Normalized-Longest Common Subsequence (GAN-LCS)

Pribadi [20] proposed the GAN-LCS method to deal with the problem of measuring the similarity of sentences with significantly different character lengths. This method uses the Longest Common Subsequence (LCS) method, which measures the length of the intersection of characters between the sentence of the reference answer (R) and the sentence of the student answer (S). The GAN-LCS method is formulated in (3), with the denominator $min(|R|,|S|)$ utilized to eliminate non-contributive words and increase the similarity between the two sentences:

$$Sim(R, S) = \frac{2\sqrt{|R||S|}}{|R| + |S|}.\exp\left(\log\left(\frac{lsc_{R \cap S}}{\min(|R|,\ |S|)}\right)\right) \tag{3}$$

Equation (3) determines the degree of similarity between the student response and each reference response. We are using four reference answers in this research; therefore, each student will get four temporary scores, which will later be converted to a final score by multiplying the maximum similarity of the four previous values by the maximum score for each question as shown in (4):

$$Score = \max\left(Sim\left(S_i, R_j\right)\right) * ms \tag{4}$$

$Sim(S_i,R_j)$ is the similarity score between the i-th student answer and the j-th reference answer. The ms is the maximum score that a student can achieve if the answer is very similar to the reference answer. In this study, we set the maximum score to 10 since this score is the highest score based on our dataset.

2.8. Performance Evaluation

In this study, the evaluation metrics used are two popular metrics used to score a continuous variable—Pearson's Correlation test and the Root Mean Square Error test [36]. The Pearson's Correlation test is used to measure the gap between human-generated and system-generated scores. This test is represented in the range from -1 to 1, where the higher value means the better result. The correlation value (r) equation is shown in (5):

$$Correlation(x, y) = \frac{\sum(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum(x_i - \overline{x})^2\ \sum(y_i - \overline{y})^2}} \tag{5}$$

The second test is the Root Mean Square Error or RMSE. This test is to measure the differences between system-generated and human-generated scores. This test is represented by the value from 0 to unlimited. This test result is always bigger than 0, and the lower the RMSE's value is, the better it performs. The RMSE equation is shown in (6):

$$RMSE = \sqrt{\frac{\sum_{n=1}^{N}(\hat{y_n} - y_i)^2}{T}} \tag{6}$$

In addition, we added a third test called Mean Absolute Percentage Error (MAPE) [37] to measure the average differences between system-generated and human-generated scores. This evaluation can also measure the success rate of ASAS [16]. The MAPE equation is shown in (7), where $x_{(i)}$ is the score from the lecturer and $y_{(i)}$ is the system-generated score, while n is the number of the data. After the mean absolute error using the MAPE equation is obtained, we can also get the Percentage Accuracy (PA) [37] by subtracting the value of 100 with the MAPE value, as shown in (8):

$$MAPE = \frac{\sum_{i=1}^{n} \left| \frac{|x_{i-} \ y_i \ |}{x_i} \right|}{n} \times 100 \tag{7}$$

$$PA = 100 - MAPE \tag{8}$$

The very last evaluation test used in this study was the Execution Time Test. This test is used to measure and compare the difference in character length before and after the proposed method is applied. The output was evaluated using Percentage Change, as shown in (9), to show the improvement in the performance in percentage value. This test also shows us the correlation to the time consumption needed to process the whole dataset in a scatter plot chart. We have also evaluated the algorithm performance through time complexity using Big O Notation.

$$Percentage\ Change = 100\ \times \frac{(final\ value - initial\ value)}{|initial\ value|} \tag{9}$$

3. Experiment Result

3.1. Abbreviations Checker

The implementation of an abbreviations checker for the reference answer (R) and the student answers ($S1$ and $S2$) with the defined dictionary (D) can be described as follows:

D = {'*dad*', '*dfd*', '*erd*', . . . '*satd*'}

R = '*dfd kamus data erd*'

*S*1 = '*dfd kamus data erd*'

*S*2 = '*data flow diagram kamus data entity relationship diagram*'

It is important to remember that both the reference answer and student answers have had the pre-processed method applied beforehand; therefore, all of the answer has been transformed into a single long sentence. At this point, we can see that R and $S1$ have a perfect match, but with $S2$, even though it has the same meaning, it has different character length and sentence structure.

The first step in applying the abbreviations checker is to find the terms in D that appeared in R and to store it as a new dictionary (D'), then extract the definition of each term, as shown in Table 3.

Table 3. Example of the used abbreviations dictionary (D').

Term	Definition
dfd	data flow diagram
erd	entity relationship diagram

The next step is to append the definition exactly after the position of the term in the sentence. For this step, if the definition has been written in the sentence and the term is not found in the sentence, the term will be inserted into the left position of the definition. In this case, we can apply this to Sentence R, $S1$, and $S2$, then the results are stored as R', $S1'$, and $S2'$:

R′ = '*dfd **data flow diagram** kamus data erd **entity relationship diagram***'

*S*1′ = '*dfd **data flow diagram** kamus data **erd entity relationship diagram***'

*S*2′ = '***dfd** data flow diagram kamus data **erd** entity relationship diagram*'

We can see the changes in each sentence marked by the word in bold. At this point, three sentences have been modified into a new form. Both $S1'$ and $S2'$ now have a perfect match to R'. This operation has been performed without changing the meaning of the sentences.

3.2. GAN-LCS Scoring

The GAN-LCS scoring can be applied directly to measure the similarity between two sentences. In this study, we created several scenarios to discover the effect of using an abbreviations dictionary to enhance GAN-LCS Scoring performance. Here is an example to calculate the similarity between two sentences using two versions of sentences, the original one and one modified by the abbreviation checker, marked by the accent symbol:

R = *'dfd kamus data erd'*
R' = *'dfd data flow diagram kamus data erd entity relationship diagram'*
$S1$ = *'dfd kamus data erd'*
$S1'$ = *'dfd data flow diagram kamus data erd entity relationship diagram'*
$S2$ = *'data flow diagram kamus data entity relationship diagram'*
$S2'$ = *'dfd data flow diagram kamus data erd entity relationship diagram'*

Scenario 1:
Sim(R, $S1$) and Sim(R, $S2$)

$$Sim(R,\ S1) = \frac{2\sqrt{20 \times 20}}{20 + 20}.exp\left(\log\left(\frac{20}{20}\right)\right) = 1$$

$$Sim(R,\ S2) = \frac{2\sqrt{20 \times 56}}{20 + 56}.exp\left(\log\left(\frac{20}{20}\right)\right) = 0.88$$

Scenario 2:
Sim(R', $S1'$) and Sim(R', $S2'$)

$$Sim(R',\ S1') = \frac{2\sqrt{64 \times 64}}{64 + 64}.exp\left(\log\left(\frac{64}{64}\right)\right) = 1$$

$$Sim(R',\ S2') = \frac{2\sqrt{64 \times 64}}{64 + 64}.exp\left(\log\left(\frac{64}{64}\right)\right) = 1$$

In Scenario 2, we can see that the $S2$ score was higher by 0.12 or increased by 12% after applying the abbreviations checker and the Scenario 2 score now has a perfect match with the human-generated score without interfering with the meaning of the sentence.

3.3. Performance Evaluation

Performance evaluation in this study focused on four outputs. The first output is the performance of the proposed method through 585 answers from students. These outputs are shown in Table 4, and the performance is measured by four metrics: Correlation, RMSE, MAPE, and PA.

Table 4. Performance Evaluation of the Proposed Method.

Question No	Response	Correlation	RSME	MAPE	PA
A1	59	0.726	1.386	14.15%	85.85%
A2	59	0.284	1.685	12.17%	87.83%
A3	59	0.692	1.086	8.33%	91.67%
A4	59	0.724	0.896	5.89%	94.11%
A5	58	0.39	1.547	13.94%	86.06%
A6	59	0.771	0.896	6.96%	93.04%
A7	59	0.208	1.701	14.25%	85.75%
A8	59	0.261	1.775	17.56%	82.44%
A9	57	0.784	1.371	15.12%	84.88%
A10	57	0.552	0.938	7.25%	92.75%
Total/Average	585	0.539	1.328	11.56%	88.44%

The second output compares the GAN-LCS performance and the proposed method performance through RMSE, correlation, and the MAPE value. These comparisons are represented by three graphs, as shown in Figure 3.

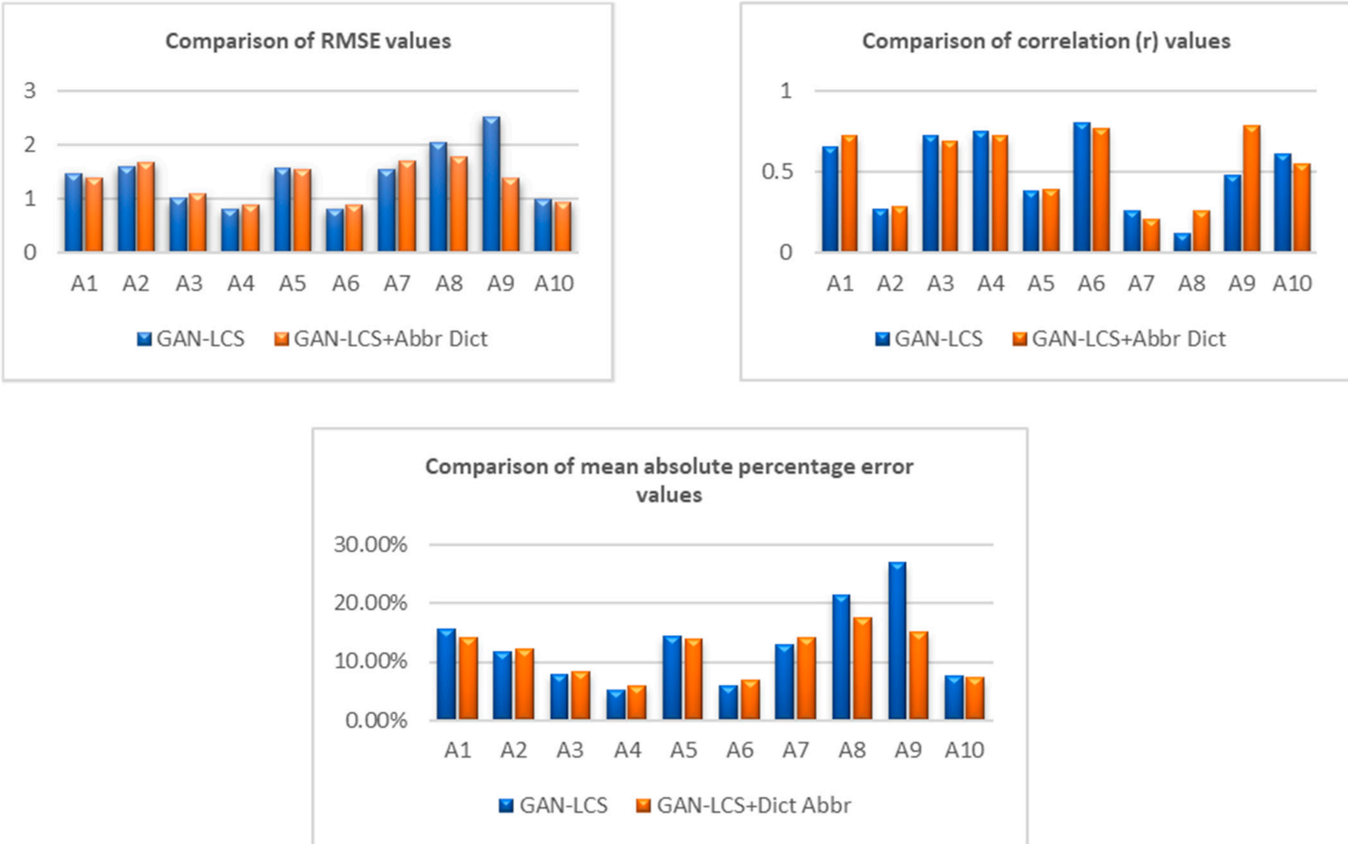

Figure 3. Comparison of RMSE, Correlation, and MAPE between original GAN-LCS and the proposed method.

The third output, as shown in Table 5, summarizes the performance comparison between the two methods regarding execution time, character length, average value of the correlation, Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and Percentage Accuracy (PA).

Table 5. Performance Comparison between GAN-LCS and the Proposed Method.

Evaluation Metric	GAN-LCS	* Proposed Method	Actual Difference	Percentage Change
Execution Time	17.41 s	10.72 s	6.69 s	34.43% faster
Character Length	114.395	93.086	21.309	18.63% less character
Correlation	0.50	0.54	0.04	8% higher
RMSE	1.438	1.328	0.110	7.65% lower
MAPE	12.94%	11.56%	1.37%	1.37% lower
PA	87.07%	88.44%	1.37%	1.37% lower

* GAN-LCS + Stemmer, Stopword Removal, and Abbreviations Checker.

The last output is the performance evaluation of the algorithm through time complexity using Big O Notation. There are two outputs in this performance evaluation, the first is the comparison of character length and time consumption between GAN-LCS and the proposed method after the stemming process, stopword, and abbreviations checker is added, as shown in Table 6. The second is a graph, as shown in Figure 4, that illustrates the comparison of total character length and total algorithm time execution.

Table 6. Performance Comparison between GAN-LCS and the Proposed Method.

Question Number	Character Length				Execution Time (in Seconds)			
	GAN-LCS	* Proposed Method	Actual Diff	Percentage Change	GAN-LCS	* Proposed Method	Actual Diff	Percentage Change
A1	7.810	**6.234**	−1.576	−20.18%	0.7127	**0.3536**	−0.3591	−50.39%
A2	14.717	**11.857**	−2.860	−19.43%	3.1934	**1.8487**	−1.3447	−42.11%
A3	10.403	**8.256**	−2.147	−20.64%	1.6016	**0.8717**	−0.7299	−45.57%
A4	14.802	**11.326**	−3.476	−23.48%	3.3080	**1.6682**	−1.6398	−49.57%
A5	21.048	**16.521**	−4.527	−21.51%	2.8806	**1.6945**	−1.1861	−41.18%
A6	10.302	**8.185**	−2.117	−20.55%	1.3898	**1.1271**	−0.2627	−18.90%
A7	9.410	**7.778**	−1.632	−17.34%	1.4551	**0.8853**	−0.5698	−39.16%
A8	10.629	**8.626**	−2.003	−18.84%	1.3876	**0.7360**	−0.6516	−46.96%
A9	**7.409**	7.499	90	+1.21%	**0.1133**	0.3851	0.2718	+239.89%
A10	9.327	**7.988**	−1.339	−14.36%	0.8315	**0.6394**	−0.1921	−23.10%
Total/Avg	115.857	**94.270**	−21.580	−18.63%	16.8736	**10.2096**	−6.6640	−39.49%

* GAN-LCS + Stemmer, Stopword Removal, and Abbreviations Checker.

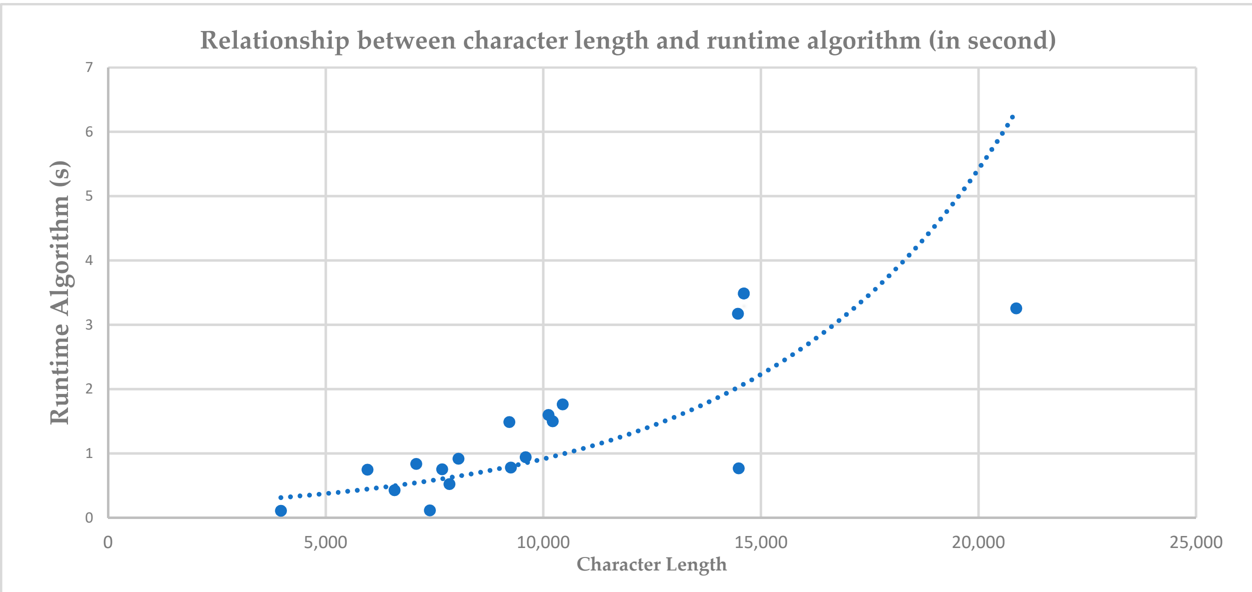

Figure 4. Abbreviations checker process.

4. Discussion

As shown in Table 4, the approach showed an overall performance with an 84.44% mean accuracy score for ten questions. The best performance was presented by the question number A4 with an accuracy score of 94.11% and 5.89% for the error mean score.

Additionally, it showed the greatest and lowest value of correlation. The lowest value of correlation, 0.208, was obtained by question number A7, while question number A9 presented 0.784 as the greatest correlation value. In the same way, the best RMSE value was 0.896 for question numbers A4 and A6, with an average of 1.328, which means this value is still better than the original method.

A significant result occurred with question number A9. The proposed method achieved 84.88% Percentage Accuracy while only 73.04% with the original method, which means there is an improvement in accuracy of 11.84%. Nevertheless, this achievement has a drawback with the execution time taking longer because of the addition of characters in the sentences, an effect of the abbreviation checker that will append the definition or the term of abbreviation inside the question. As a result, the scoring process becomes slower, 0.27 s from the original method, as shown in Table 5.

The GAN-LCS time performance was dependent on the total character number of the sentences. Figure 4 illustrates that the time execution required increased following the exponential line trend. This is because the GAN-LCS has time complexity *O(m.n)*, where *m* is the character length of the reference answer, and *n* is the character length of the student answer. Since there are four reference answers, the time required for scoring one answer is then multiplied by four.

The time complexity *O(m.n)* was completely influenced by LCS time complexity since it has the most significant effect on the whole process in (3). Based on this, the implementation of stemming and stopword removal will help reduce the non-contributive characters in the sentence so that the process will be a lot faster.

There are two contributions in this study. The first is the proposed method named the abbreviations checker to minimalize the character length difference and increase the character intersection between two sentences without changing the meaning of the sentence. This method will restructure the sentence so that both the reference answer and student answer get an additional term that will expand the query used to retrieve more relevant results.

However, the abbreviations dictionary generation used in this research still used the manual approach due to the possibility that an abbreviation has a different meaning in other domains.

The second is the implementation of stemming and stopword removal in the pre-processing process. These two pre-processing stages reduced the number of characters by 21,309 characters or approximately 18.63%, as shown in Table 5. This reduction also increases the number of the same characters and the same word structure throughout the stemming process. As a result, besides reducing execution time, this method can also slightly increase the accuracy. This method reduces the time execution by 6.69 s or 34.43% compared to the original one.

Overall, the system has good performance, although it still has a shortcoming in terms of low correlation found in questions A2, A5, A7, and A8. It occurred because questions A2 and A5 were a type of open-answer question that required a longer length response, while questions A7 and A8 demanded the students explain certain steps in the correct order. This is because the method used does not look at the sentence according to its language structure but only sees the sentence through the character structure that composes it. However, this approach has the advantage of being able to be used widely without being limited by the language.

Additionally, this study also showed how the students answered the question. There were some students answered the question by incorporating the question into their response while others did not. These ways of how students answered the questions influenced the final score. Therefore, to overcome that problem, it needed to incorporate a method to remove the question sentence from the answers.

5. Conclusions

Based on the result of this study, the approaches provide an overall improvement to the GAN-LCS method. It showed a 0.110 lower value in RMSE and 0.04 higher value in correlation, or an increased by 8%. In addition, the MAPE values were lower by 1.37%, which meant the accuracy was increased by 1.37%. Moreover, this approach was able to fasten the execution duration by 34.43%.

Further research is expected to be able to process the dataset with a more varied question type to handle other ways students answer, especially for short answer type questions. The automatic approach for generating an abbreviation dictionary based on the exam session is also expected to be developed with future research.

In the future, the accuracy of GAN-LCS can be improved by using other query expansion techniques, such as synonymous and polysemous words to retrieve relevant information so that it can identify sentences with the same meaning but different structures.

Author Contributions: Conceptualization, A.-R.M. and I.H.; methodology, A.-R.M.; validation, A.-R.M., A.E.P. and I.H.; formal analysis, A.-R.M., A.E.P. and I.H.; investigation, A.-R.M.; resources, A.-R.M.; data curation, A.-R.M.; writing—original draft preparation, A.-R.M.; writing—review and editing, A.-R.M., A.E.P. and I.H; visualization, A.-R.M.; supervision, I.H and A.E.P.; project administration, A.-R.M.; funding acquisition, A.-R.M. and I.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Politeknik Negeri Ketapang, grant number 224/PL39/KP/BP/2022. The APC was funded by the thesis recognition program (RTA) from Universitas Gadjah Mada, grant number 3550/UN1.P.III/Dit-Lit/PT.01.05/2022.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AES	Automatic Essay Scoring
ASAS	Automatic Short Answer Scoring
DFD	Data Flow Diagram
ERD	Entity Relationship Diagram
GAN-LCS	Geometric Average Normalized-Longest Common Subsequence
LCS	Longest Common Subsequence
LSA	Latent Semantic Analysis
MAPE	Mean Absolute Percentage Error
MMR	Maximum Marginal Relevance
PA	Percentage Accuracy
QE	Query Expansion
RMSE	Root Mean Square Error
SATD	Structured Analysis and Design Technique
STS	Semantic Text Similarity

References

1. Hebebci, M.T.; Bertiz, Y.; Alan, S. Investigation of Views of Students and Teachers on Distance Education Practices during the Coronavirus (COVID-19) Pandemic. *Int. J. Technol. Educ. Sci.* **2020**, *4*, 267–282. [CrossRef]
2. Pregowska, A.; Masztalerz, K.; Garlińska, M.; Osial, M. A Worldwide Journey through Distance Education—From the Post Office to Virtual, Augmented and Mixed Realities, and Education during the COVID-19 Pandemic. *Educ. Sci.* **2021**, *11*, 118. [CrossRef]
3. Hoofman, J.; Secord, E. The Effect of COVID-19 on Education. *Pediatr. Clin. N. Am.* **2021**, *68*, 1071–1079. [CrossRef] [PubMed]
4. Impey, C.; Formanek, M. MOOCS and 100 Days of COVID: Enrollment surges in massive open online astronomy classes during the coronavirus pandemic. *Soc. Sci. Humanit. Open* **2021**, *4*, 100177. [CrossRef] [PubMed]
5. Burrows, S.; Gurevych, I.; Stein, B. The Eras and Trends of Automatic Short Answer Grading. *Int. J. Artif. Intell. Educ.* **2015**, *25*, 60–117. [CrossRef]
6. Perez, D.; Alfonseca, E. Application of the Bleu algorithm for recognising textual entailments. In Proceedings of the First Challenge Workshop, Southampton, UK, 11–13 April 2005.
7. Xi, Y.; Liang, W. Automated Computer-Based CET4 Essay Scoring System. In Proceedings of the 2011 Third Pacific-Asia Con-ference on Circuits, Communications and System (PACCS), Wuhan, China, 17–18 July 2011. [CrossRef]
8. Siddiqi, R.; Harrison, C.J.; Siddiqi, R. Improving Teaching and Learning through Automated Short-Answer Marking. *IEEE Trans. Learn. Technol.* **2010**, *3*, 237–249. [CrossRef]
9. Lahitani, A.R.; Permanasari, A.E.; Setiawan, N.A. Cosine similarity to determine similarity measure: Study case in online essay assessment. In Proceedings of the 2016 4th International Conference on Cyber and IT Service Management, Bandung, Indonesia, 26–27 April 2016; pp. 1–6. [CrossRef]

10. Fitri, R.; Asyikin, A.N. Aplikasi penilaian ujian essay otomatis menggunakan metode cosine similarity. *J. Poros Tek.* **2015**, *7*, 54–105. [CrossRef]
11. Fauzi, M.A.; Utomo, D.C.; Setiawan, B.D.; Pramukantoro, E.S. 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, Bangkok, Thailand, 25–27 August 2017. [CrossRef]
12. Hasanah, U.; Permanasari, A.E.; Kusumawardani, S.S.; Pribadi, F.S. A scoring rubric for automatic short answer grading system. *TELKOMNIKA* **2019**, *17*, 763. [CrossRef]
13. Aji, R.B.; Baizar, A.; Firdaus, Y. Automatic essay grading system menggunakan metode latent semantic analysis. In Proceedings of the Seminar Nasional Aplikasi Teknologi Informasi (SNATI), Yogyakarta, Indonesia, 17–18 June 2011.
14. Yustiana, D. Penilaian otomatis terhadap jawaban esai pada soal berbahasa indonesia menggunakan latent semantic analysis. In Proceedings of the Seminar Nasional Inovasi dalam Desain dan Teknologi, Surabaya, Indonesia, 19 March 2015.
15. Ratna, A.A.P.; Budiardjo, B.; Hartanto, D. SIMPLE: Sistim penilai esei otomatis untuk menilai ujian dalam bahasa Indonesia. *Makara J. Technol.* **2010**, *11*, 5–11. [CrossRef]
16. Citawan, R.S.; Mawardi, V.C.; Mulyawan, B. Automatic Essay Scoring in E-learning System Using LSA Method with N-Gram Feature for Bahasa Indonesia. In Proceedings of the 3rd International Conference on Electrical Systems, Technology and Information (ICESTI 2017), Bali, Indonesia, 26–29 September 2017. [CrossRef]
17. Amalia, A.; Gunawan, D.; Fithri, Y.; Aulia, I. Automated Bahasa Indonesia essay evaluation with latent semantic analysis. In Proceedings of the 3rd International Conference on Computing and Applied Informatics 2018, Medan, Indonesia, 18–19 September 2019. [CrossRef]
18. Astutik, S.; Cahyani, A.D.; Sophan, M.K. Sistem penilaian esai otomatis pada e-learning dengan algoritma winnowing. *J. Inform.* **2014**, *12*, 47–52. [CrossRef]
19. Hasanah, U.; Hartato, B.P. Assessing Short Answers in Indonesian Using Semantic Text Similarity Method and Dynamic Corpus. In Proceedings of the 2020 12th International Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta, Indonesia, 6–8 October 2020; pp. 312–316. [CrossRef]
20. Pribadi, F.S.; Permanasari, A.E.; Adji, T.B. Short answer scoring system using automatic reference answer generation and geometric average normalized-longest common subsequence (GAN-LCS). *Educ. Inf. Technol.* **2018**, *23*, 2855–2866. [CrossRef]
21. Manning, C.D.; Raghavan, P.; Schuetze, H. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008.
22. Tala, F.Z. A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia. Master's Thesis, Universiteit van Amsterdam, Amsterdam, The Netherlands, 2003.
23. Stopwords Indonesian (ID). Available online: https://github.com/stopwords-iso/stopwords-id (accessed on 30 June 2022).
24. Carvalho, G.; de Matos, D.M.; Rocio, V. Document Retrieval for Question Answering: A Quantitative Evaluation of Text Preprocessing. In Proceedings of the First Ph.D. Workshop in CIKM, PIKM 2007, Sixteenth ACM Conference on Information and Knowledge Management, Lisbon, Portugal, 9 November 2007. [CrossRef]
25. Sastrawi. Available online: https://github.com/sastrawi/sastrawi (accessed on 30 June 2022).
26. Adriani, M.; Asian, J.; Nazief, B.; Tahaghoghi, S.M.M.; Williams, H.E. Stemming Indonesian: A confix-stripping approach. *ACM Trans. Asian Lang. Inf. Process.* **2007**, *6*, 1–33. [CrossRef]
27. Asian, J. Effective Techniques for Indonesian Text Retrieval. Ph.D. Thesis, RMIT University, Melbourne, Australia, 2007.
28. Arifin, A.Z. Enhanced confix stripping stemmer and ants algorithm for classifying news document in indonesian language. In Proceedings of the 5th International Conference on Information & Communication Technology and Systems (ICTS), Surabaya, Indonesia, 4 August 2009.
29. Tahitoe, A.D.; Purwitasari, D. Implementasi Modifikasi Enhanced Confix Stripping Stemmer untuk Bahasa Indonesia dengan Metode Corpus Based Stemming. Bachelor's Thesis, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia, 2010.
30. Azad, H.K.; Deepak, A. Query expansion techniques for information retrieval: A survey. *Inf. Process. Manag.* **2019**, *56*, 1698–1735. [CrossRef]
31. Noorbehbahani, F.; Kardan, A.A. The automatic assessment of free text answers using a modified BLEU algorithm. *Comput. Educ.* **2011**, *56*, 337–345. [CrossRef]
32. Rodrigues, F.; Araújo, L. Automatic assessment of short free text answers. In Proceedings of the 4th International Conference on Computer Supported Education, Porto, Portugal, 16–18 April 2012. [CrossRef]
33. Leacock, C.; Chodorow, M. C-rater: Automated Scoring of Short-Answer Questions. *Comput. Humanit.* **2003**, *37*, 389–405. [CrossRef]
34. Mohler, M.; Mihalcea, R. Text-to-text semantic similarity for automatic short answer grading. In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics on—EACL-09, Athens, Greece, 30 March–3 April 2009. [CrossRef]
35. Carbonell, J.; Goldstein, J. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval—SIGIR-98, Melbourne, Australia, 24–28 August 1998. [CrossRef]
36. Thakkar, M. *Finetuning Transformer Models to Build ASAG System*; University of Limerick: Limerick, Ireland, 2021.
37. Ko, Y.; Han, S. A Duration Prediction Using a Material-Based Progress Management Methodology for Construction Operation Plans. *Sustainability* **2017**, *9*, 635. [CrossRef]