IJCCS (Indonesian Journal of Computing and Cybernetics Systems)
Vol.16, No.1, January 2022, pp. 79~90
ISSN (print): 1978-1520, ISSN (online): 2460-7258
DOI: https://doi.org/10.22146/ijccs.69906

The Effect of Text Summarization in Essay Scoring (Case Study: Teach on E-Learning)

Sensa Gudya Sauma Syahra[*1], Yunita Sari[2], Yohanes Suyanto[3]
[1]Master Program of Computer Science, FMIPA UGM, Yogyakarta, Indonesia
[2,3]Department of Computer Science and Electronics, FMIPA UGM, Yogyakarta, Indonesia
e-mail: [*1]**sensa.gudya@mail.ugm.ac.id**, [2]yunita.sari@ugm.ac.id, [3]yanto@ugm.ac.id

Abstrak

Perkembangan automated essay scoring (AES) dalam pendekatan neural network (NN) telah meniadakan feature engineering. Akan tetapi, feature engineering masih dibutuhkan, terlebih lagi data dengan label berupa nilai rubrik, yaitu pelengkap nilai holistik AES, masih jarang ditemukan. Secara umum, data tanpa label/nilai lebih banyak ditemukan. Namun, penelitian unsupervised AES tidak berkembang dengan penggunaan data publik berlabel yang lebih umum. Berdasarkan studi kasus yang diangkat dalam penelitian, peringkasan teks otomatis (ATS) digunakan sebagai model feature engineering AES dan readability index sebagai definisi nilai rubrik untuk data tanpa label.

Penelitian ini berfokus pada pengembangan AES dengan mengimplementasikan hasil ATS pada SOM dan HDBSCAN. Data yang digunakan dalam penelitian merupakan data esai TEACH ON E-learning sebanyak 403 dokumen. Data direpresentasikan dalam bentuk kombinasi vektor kata dan readability index. Berdasarkan pengujian dan pengukuran yang dilakukan, disimpulkan bahwa AES dengan implementasi ATS belum berpotensi baik untuk penilaian esai TEACH ON dalam meningkatkan silhouette score. Model tersebut menghasilkan silhouette score terbaik sebesar 0.727286113 dengan data esai asli.

Kata kunci— *AES, ATS, readability index, unsupervised*

Abstract

The development of automated essay scoring (AES) in the neural network (NN) approach has eliminated feature engineering. However, feature engineering is still needed, moreover, data with labels in the form of rubric scores, which are complementary to AES holistic scores, are still rarely found. In general, data without labels/scores is found more. However, unsupervised AES research has not progressed with the more common use of publicly labeled data. Based on the case studies adopted in the research, automatic text summarization (ATS) was used as a feature engineering model of AES and readability index as the definition of rubric values for data without labels.

This research focuses on developing AES by implementing ATS results on SOM and HDBSCAN. The data used in this research are 403 documents of TEACH ON E-learning essays. Data is represented in the form of a combination of word vectors and a readability index. Based on the tests and measurements carried out, it was concluded that AES with ATS implementation had no good potential for the assessment of TEACH ON essays in increasing the silhouette score. The model produces the best silhouette score of 0.727286113 with original essay data.

Keywords— *AES, ATS, readability index, unsupervised*

1. INTRODUCTION

The automated essay scoring (AES) modelling has developed in the neural network (NN) approach and has succeeded in achieving its state-of-the-art in supervised learning. Problems arise when AES is more developed by only applying holistic values, namely a final score that covers

Received October 22th,2021; Revised January 31th, 2022; Accepted January 31th, 2022

the entire assessment process, while this value cannot be fully used to determine the quality of an essay [1]. The need to define the quality of the essay encourages the development of AES using rubric values as an initial explanation layer in defining and completing holistic values [2]. Readability contains the ease with which the reader understands a document or text because of its writing style [3]. The concept of readability can be used as an approach to understanding essays through the choice of words, phrases, sentences, and how the elements are arranged, where the understanding process is one of the domain definitions for determining essay quality (rubric scores). [4] utilize a readability index in understanding inter-language words using lexical analysis.

The ever-evolving NN approach in AES modelling has eliminated the feature engineering process. However, rubric values cannot be defined only by NN modelling. In addition, data (corpora) with labels in the form of rubric values in large quantities is still rarely found. This condition makes feature engineering still needed in AES. [5] and [6] also mention that an effective and precise feature engineering process is no less important in AES. The importance of feature engineering in AES can be met with the automatic text summarization (ATS) approach. This is reinforced by the statement of Dong and Zhang (2016) which states that summarization is a civilized process in essay assessment. Features can be extracted well by ATS because documents can be analysed in a concise form and consist of important semantic parts.

TEACH ON which is used as research case study data is an essay that does not have a value reference, both holistic and rubric. This condition shows the TEACH ON essay as unlabelled data. The general form of learning with these data conditions can be done with unsupervised learning. Chen et al. (2010) started research on AES using unsupervised/weakly-supervised learning. However, AES in the form of unsupervised is no longer developing because AES research uses more labelled corpora, such as the Automated Student Assessment Prize (ASAP). AES modelling with ASAP is not appropriate for evaluating TEACH ON essays. The difference in the characteristics of the two data makes TEACH ON essays need to be processed in a different modelling from modelling with ASAP.

Based on the TEACH ON case study, this research will conduct AES modelling with unsupervised learning that applies ATS at an early stage. The AES model developed will be evaluated based on 2 TEACH ON assessment criteria, namely completeness and cohesion of ideas. The combination of word vectors and readability index is used as a representation of the assessment data in the AES model. This study focuses on how the effect of text summarization in essay assessment is in the form of unsupervised.

2. METHODS

In this section, the proposed method is explained in detail. This includes the data used in this research, pre-processing and representing the data, and the ATS-AES model.

2.1 Data Analysis

The data are reflective essays collected by institutions in English. The reflective essay was written by the teacher after watching an inspirational video from MOOC. The data consists of 403 essays with content containing at least 800 words. The resulting essay must meet 3 main contents, which are as follows:

1. Explain why passion arises in relation to passion as a teacher.
2. Describe the author's vision of the future and how he is remembered as a teacher by students.
3. Identify and discuss 3 actions, steps, and changes you would like to make to achieve the vision.

Essays are analysed early based on many sentences and words. The analysis process is done by counting the number of words and sentences. Based on the calculations performed, the results are obtained as shown in Table 1.

The assessment criteria are divided into 2 parts. First, the assessment of completeness. This assessment evaluates the completeness of the content of reflective essays based on 3 main contents. Second, the assessment of cohesion of ideas. This assessment evaluates how the essay is written, by looking at the grammar and writing structure used.

Table 1 Word and sentence analysis of essays

	Word	Sentence
Max	2241 (406.txt)	146 (186.txt)
Min	272 (99.txt)	11 (248.txt)

This study uses several forms of data for the evaluation and testing process of essay assessment modelling. The forms of data used in the study are original essay documents, readability index feature data, summarized documents, and data from combinations of original/summary essays and readability index.

2.2 Architecture Model

In general, the construction of an automated essay assessment model carried out in research is shown in the form of a flow chart in Figure 1.

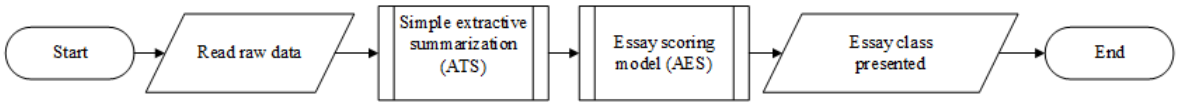

Figure 2 General research flowchart

The research carried out 2 main phases, namely the essay summary phase (ATS) and the AES itself. ATS is carried out in the form of unsupervised learning in the form of extractive summarization. The ATS model results are subjected to ROUGE evaluation before being taken as an AES modelling input document. Essay summary is processed in AES modelling. With clustering modelling, essays are grouped into appropriate clusters. The formed cluster is evaluated by measuring the silhouette coefficient.

2.3 Data Pre-processing

Pre-processing in this study aims to process a collection of documents to be used as input in the form of a predetermined data representation. It consists of case-folding, filtering, stop-word elimination, tokenization, and lemmatization. Document pre-processing is carried out twice, namely in the summary and assessment model at the same stage. The complete pre-processing flow chart is shown in Figure 2 and an example of the result is shown in Table 2.

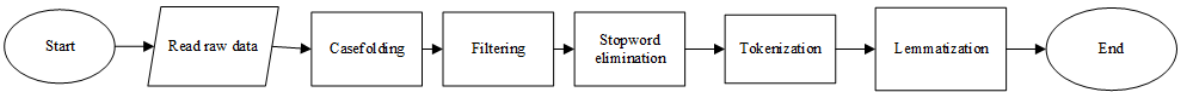

Figure 3 Data pre-processing

Table 2 Sentence pre-processing results

Sentence	Step	Result
Since the time I was in process of learning my work up to the time I am already applying it, I always hear experts say that "learning is a continuous process".	Case-folding	since the time I was in process of learning my work up to the time I am already applying it, I always hear experts say that "learning is a continuous process".
	Filtering (+ Stop-word elimination)	since time process learning work time already applying always hear experts say learning continuous process
	Tokenization	[since] [time] [process] [learning] [work] [time] [already] [applying] [always] [hear] [experts] [say] [learning] [continuous] [process]
	Lemmatization	[since] [time] [process] [learning] [work] [time] [already] [applying] [always] [hear] [experts] [say] [learning] [continuous] [process]

2.4 Data Representation

This study forms the summary data in two forms of representation based on two essay assessment criteria. The data representation for the completeness assessment criteria implements word embedding. The data representation for the cohesion of ideas assessment criteria implements a readability index.

2.4.1 Readability index

Readability index was chosen as a structured assessment because it is able to measure the reader's understanding of a text based on its writing style. In [3], readability is also defined as the extent to which a person in a certain class is able to find certain, interesting, and understandable reading material. Thus, the readability index uses formulas to predict and measure the level of understandability of a text based on certain readers.

Readability index calculation is done by several formulas. This study uses five readability index formulas to represent essays. The five formulas are Flesch's Reading Ease described in [7], and Flesch-Kincaid Grade Reading Level, Gunning Fog Index, Automated Readability Index, and Coleman-Liau Index described in [3]. These five values were chosen as general calculations for measurements with a readability index. The formulas are as follows.

1. Flesch's Reading Ease
   $$FRE = 206{,}835 - (1{,}015 \times ASL) - (84{,}6 \times ASW) \quad (1)$$
   Formula 1 shows Flesch's Reading Ease calculations. The FRE range is in the value 0 – 100, with FRE = 30 indicating the document is very difficult to read and FRE = 70 being a document that is suitable for reading for adults. ASW is the average number of syllables-per-word, the number of syllables divided by the number of words. ASL is the average sentence length, which is the number of words divided by the number of sentences.
2. Flesch-Kincaid Grade Reading Level
   $$FK = 0{,}39(TW/TotSent) + 11{,}8((TotSyll)/(TW)) - 15{,}59 \quad (2)$$
   Formula 2 shows the calculation of the Flesch-Kincaid Grade Reading Level. The calculation is done by analysing the various types of words that appear in the document. Abbreviations (example: don't), strings of digits (example: 1,000,000), and conjunctions (example: second-grade) count as 1 letter. Grammatically unrelated sentences or clauses are considered as separate sentences or clauses from each other. For syllables, the digits can be counted as many as the words for the pronunciation of the digits. TW is the total words, TotSent is the total sentences, and TotSyll is the total syllables.
3. Gunning Fog Index
   $$GF = 0{,}4\left[\left(\frac{words}{sentences}\right) + 100\left(\frac{complex\ words}{words}\right)\right] \quad (3)$$
   In formula 3, there are two statistical factors to predict readability, namely sentence length and how difficult a word is. Measurement of a word is considered difficult by looking at the number of syllables consisting of three or more syllables. However, names, combinations of short words, or verb forms (which can be three or more syllables with the addition of -ed, -es) are not counted as complex words. Documents are easy to read when the fog value is low (±5) and more difficult when the fog value is higher (±20).
4. Automated Readability Index
   $$ARI = 4{,}71\left(\frac{characters}{words}\right) + 0{,}5\left(\frac{words}{sentences}\right) - 21{,}43 \quad (4)$$
   Formula 4 displays the Automated Readability Index calculation. In this formula, the 2 determining factors are the number of characters-per-word and word-per-sentence. However, documents also have conjunctions, dashes, and other elements to take into account. Thus, this equation can lead to inconsistency of interpretation.
5. Coleman-Liau Index
   $$CL = 0{,}0588L - 0{,}296S - 15{,}8 \quad (5)$$
   Formula 5 displays the readability index calculation with the Coleman-Liau Index. L is the average number of letters per-100 words and S is the average number of sentences

per-100 words. What makes this equation ambiguous is how the sentences for the equation are calculated.

2.4.2 Word embedding

The construction of word vectors is carried out within the scope of the essay document vocabulary that has been subjected to pre-processing. This is done to study the relationship of words in the data with the appropriate context and not limited to the use of vectors that are already available in the pre-trained word embedding.

The word embedding model used in the research for AES modelling is doc2vec with a form of self-learning. Enter the doc2vec model consisting of word vectors and paragraph vectors. Each word vector is a token in the vocabulary which is represented in a 1×V one-hot vector. V is the number of tokens in the vocabulary based on the form of data used. The tokens in the vocabulary are 163982 for the original essay data, 77888 for the TF-IDF summary, and 78378 for the TextRank summary. Paragraph vector or document id has dimensions of 1×D, where D is the number of documents as modelled data. The value of D in this study is 403. The weight matrix of hidden layer W is V×N, while the weight matrix of C is D×N. N is a feature vector in the form of a defined vector input value. The definition of the input vector values used in this study are 32, 64, 128, and 256. The document will be defined into the input vector size as the final result of document representation and become the input for the essay assessment model.

Doc2vec has 2 embedding models, namely distributed memory (PV-DM) and distributed bag-of-word (PV-DBOW). Conceptually, PV-DM has better performance than PV-DBOW when doing independent learning [8]. Figure 3 shows the learning framework of the two doc2vec models.

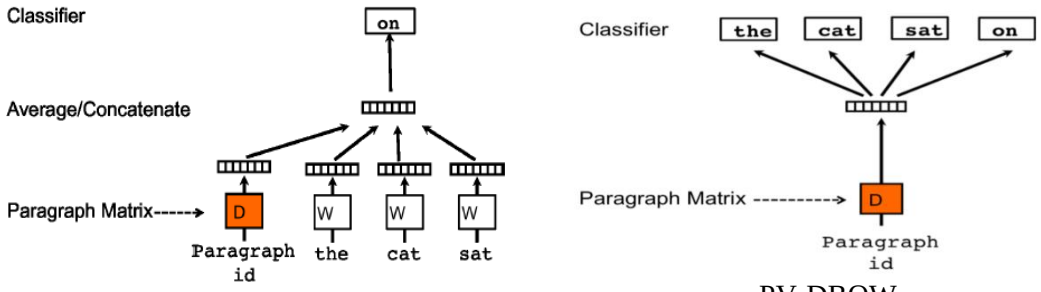

Figure 3 Doc2vec model embedding

2.5 ATS Architecture

Automatic text summarization (ATS) is carried out in an extractive form. This is because the summary results are assumed not to change the choice of words that have been made by the author. The ATS model implemented in this study consists of 2 forms, namely TF-IDF and TextRank.

2.5.1 TF-IDF Summarization

The TF-IDF summary process used in this study uses the steps defined by [9]. These steps are described in the formulation of the following formula.

1. Document indexing
   This process is carried out to determine the index term (t) which is used as a document representation. All pre-processed words are used as index terms.
2. Term weighting
   The value of each term is calculated by calculating the frequency of occurrence of the term in the document (d).
3. Long-frequency weighting

This process calculates the weight of the results of the appearance of the term weighting stage. The formula used is equation 6.

$$Wtf_{t,d} = \begin{cases} 1 + log_{10} tf_{t,d}, jika\ tf_{t,d} > 0 \\ 0, lainnya \end{cases} \quad (6)$$

where $Wtf_{t,d}$ is the log-frequency weighting on the t-th term, the d-th document and $tf_{t,d}$ is the weighting of the t-term, d-th document

4. Document frequency
   This step counts the number of documents containing the term at the t-th index.
5. Inverse document frequency
   This process is carried out to calculate the inverse value of the document frequency stage with equation 7.
   $$idf_t = log_{10} \frac{N}{df_t} \quad (7)$$
   where idf_t is the inverse document frequency on the t-term, N is the total number of existing documents, and df_t is the value of the document frequency on the t-term.
6. TF-IDF
   Calculate the value or weight of each term against the document with equation 8.
   $$W_{t,d} = Wtf_{t,d} \times idf_t \quad (8)$$
   where $W_{t,d}$ is TF-IDF on term t, document d, $Wtf_{t,d}$ is log-frequency weighting on term t, document-d, and idf_t is the inverse document frequency on term t.
7. The process for calculating the final value of each document with equation 9.
   $$Ws_j = \sum_{i=1}^{Nterm} Wtd_{i,j} \quad (9)$$
   where Ws_j is the value of the j-th document, term is the number of terms used, and $Wtd_{i,j}$ is the TF-IDF value in the i-th term, d-document.

2.5.2 TextRank Summarization

Based on the graph algorithm, TextRank can be implemented to construct extractive summaries as follows [10].

1. Define the graph that is built so that it can represent the relationships that occur in the text.
2. Define a relation that determines the relationship between two sentences when there is a resemblance.
3. The relationship that occurs can be given a weighted (weighted graph).
4. Sort the sentences based on the vertex value obtained.

The similarity between the two sentences (step 2) is measured as a function of similar contexts. This idea gave rise to a recommendation (voting) process. Recommendation occurs when a sentence that refers to a context, helps the reader to refer to other sentences in the text that have the same context so that a link can be built between the two sentences. The similarity function between the two sentences used in this study is cosine distance.

2.6 AES Architecture

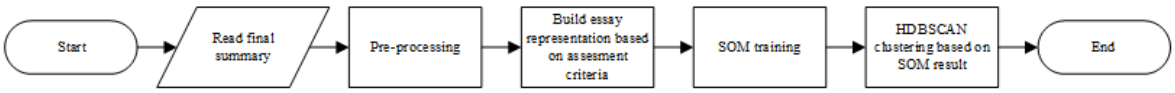

Figure 4 AES framework

Figure 4 shows a flow diagram of the unsupervised AES process using SOM and HDBSCAN. SOM studies data in its early stages. The learning outcomes, namely mapping nodes, are used as input HDBSCAN to determine the document class.

SOM is known as kohonen map which is used as unsupervised learning in NN. SOM works by grouping (clustering) based on the pattern of the data studied. In general, SOM is implemented as a feature detection. The SOM algorithm serves to convert complex high-

dimensional input into a discrete output space with lower dimensions (usually two-dimensional). SOM consists of three processes, namely competition, cooperation, and adaptation. The three main processes of SOM can be defined in the following form [11].

Figure 5 shows an illustration of the SOM architecture with a kohonen layer measuring 6x7, so there are 42 nodes in a kohonen layer. This refers to [12], which states that there is a theory to determine the number of nodes in the coherent layer, which is at least 10% of the data.

Algorithm 1 Self-Organizing Map Algorithm

```
procedure SOM(trainData):
    read trainData
    initialize weight by randomly selecting elements from trainData
    repeat
        obtain dataPoint from trainData
        find bmu of dataPoint
        determine neighbor neurons close to the bmu
        migrate neighbor neurons towards dataPoint
        update learning factor and neighborhood radius
    until pre-specified number of iterations are performed
```

After the learning process using SOM, the results that have been formed are used in further clustering modelling. The clustering modelling used utilizes a machine learning algorithm in the form of HDBSCAN. HDBSCAN can simplify the analysis needs of the SOM modelling results. The SOM learning outcomes nodes are used as input data for the HDBSCAN model. The results of HDBSCAN clustering are the final result of grouping essay documents as a form of essay assessment in this study. The HDBSCAN working process is shown in algorithm 2 [13].

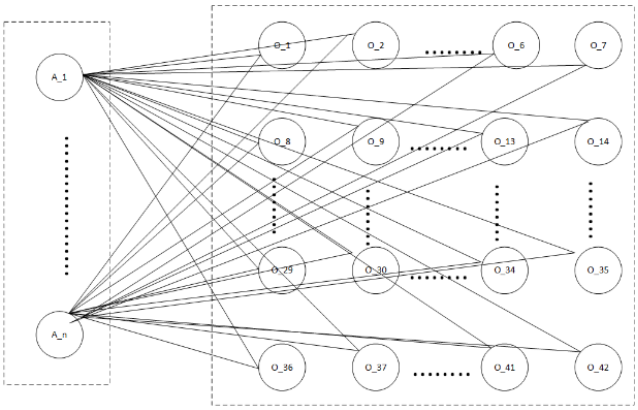

Figure 5 SOM Architecture

Algorithm 2 HDBSCAN Algorithm

```
procedure SOM(X):
    compute the core distance for all data objects in X
    compute an MST of the mutual reachability graph
    extend the MST by adding each vertex a "self-edge" with the core distance of
    the corresponding object as weight
    extract the hdbscan hierarchy as a dendogram from extended MST:
    for the root of the tree assign all objects the same label (single cluster)
    iteratively remove all edges from extended MST in decreasing order of
    weights:
      before each removal, set the dendogram scale value of the current hierarchical
      level as the weight of the edge(s) to be removed
      after each removal, assign labels to the connected component(s) that
      contain(s) the end vertex(-ices) of the removed edge(s), to obtain the next
      hierarchical level: assign new cluster label to component if it still has at least
      one edge, else assign it a null label (noise).
```

3. RESULTS AND DISCUSSION

The results of the study were obtained by evaluating the implemented summary and assessment model. The evaluation process uses variations of several parameter values so that there are parameters with fixed and changing values at this stage.

3.1 Evaluation of Summarization Model

This study uses two summary models, where the results of one model are used as a summary of the system and the other model produces a summary of targets. Both forms of summary define the summary result with 50% compression of the original. Since the research data does not have a target summary, the quality test of the summary model is based on the implementation of two forms of system summarization. Each is defined in two roles and compared. The comparison measurement uses ROUGE with the definition of the F-score value. Table 3 shows the details of the model pairs formed with the variation of the tested parameters and the average F-score results with the ROUGE variation.

The highest average value is obtained by Model 1, namely the pair TF-IDF with pos-tagging and TextRank with word2vec in the form of CBOW, amounting to 0.622184201. Thus, the summary pair of TF-IDF with pos-tagging and TextRank with word2vec in the form of CBOW was chosen to be studied in the selected AES model.

Table 3 Model couple F-score

Model	Pos Tagging TF-IDF	Word Embedding TextRank	Average F-score
Model 1	**Pos Tagging used**	**CBOW (vec_size = 32)**	**0.622184201**
Model 2		Skip-Gram (vec_size = 32)	0.617555719
Model 3	No Pos Tagging	CBOW (vec_size = 32)	0.587515568
Model 4		Skip-Gram (vec_size = 32)	0.588775622

3.2 Evaluation of Scoring Model

Assessment model testing was conducted by SOM model evaluation and HDBSCAN for clustering process.

3.2.1 Evaluation of SOM model

SOM modelling testing begins by conducting parameter experiments on doc2vec. Tests are carried out to find the best doc2vec parameters that can be implemented further into SOM modelling. Table 4 shows the test results in terms of quantization error from SOM modelling.

Table 4 Quantization error testing doc2vec

vec_size	max_epoch	DM	DBOW
32	25	0.608117735	0.735370273
64		0.957491681	1.102251321
128		1.434934009	1.732029314
256		2.009976286	2.400354566
32	**50**	**0.599938107**	0.711434881
64		0.936729613	1.116752832
128		1.512306404	1.63461923
256		2.228182094	2.604733197
32	100	1.37948655	1.348239844
64		2.105735036	2.120817675
128		3.152561992	3.191621684
256		4.675485502	4.626341941

Doc2vec DM displays the lowest error value of 0.599938107, while DBOW is 0.711434881. Both values are obtained with vec_size = 32 and max_epoch = 50. Doc2vec DM with vec_size = 32 and max_epoch = 50 is an appropriate model for the data form used in the study.

The next SOM modelling test is to experiment with SOM parameters after getting the best doc2vec modelling. Tests are carried out to find the best SOM parameters that can be implemented further into the AES modelling in the form of clustering. SOM displays the lowest error value for each number of iterations is 0.713475303, 0.637374763, 0.629293621. Two error values are obtained from the SOM model with map_size = 6x7, while the last error value is map_size = 7x6. The SOM with map_size = 7x6 has the lowest error value of all learning models. Based on the results obtained, it is shown that the essay data is suitable for studying SOM with map_size = 7x6 and 100000 iterations. Table 5 shows the test results in terms of the quantization error of SOM modelling.

Table 5 Quantization error SOM testing

map_size	Iteration	Quantization error
4x10	100	0.717600873
5x8		0.725003888
6x7		0.713475303
7x6		0.716554581
8x5		0.740875622
10x4		0.731284536
4x10	10000	0.643914539
5x8		0.64794211
6x7		0.637374763
7x6		0.643991145
8x5		0.652076373
10x4		0.645517476
4x10	**100000**	0.631562129
5x8		0.631644323
6x7		0.633854359
7x6		**0.629293621**
8x5		0.633919634
10x4		0.634873702

3.2.2 Evaluation of Clustering Model

This stage of testing implements the SOM model from the previous tests. This stage of testing performs the search for HDBSCAN parameter values that provide the best silhouette values. Tests are carried out to find the best HDBSCAN parameters which can be implemented further as AES modelling for TF-IDF and TextRank summary data. Table 6 shows the results of the silhouette score evaluation of the HDBSCAN model.

Table 6 Silhouette score HDBSCAN model

min_sample	min_cluster_size	Silhouette score
1	**10**	0.625323772
3		0.655720048
5		**0.656087932**
1	30	0.343904506
3		0.371590676
5		0.402103183
1	60	0.256669074
3		0.325796796
5		0.360078281

Based on table 6, it is found that the smaller the min_sample value, the greater the min_cluster_size value, the lower the silhouette score generated by the model. This can be because the feature values in the combination of the original essay data and the readability index

are more suitable to be collected in small classes. The best model is obtained with min_sample = 5 and min_cluster_size = 10. The model produces a silhouette score of 0.656087932. The selected HDBSCAN parameter value becomes the HDBSCAN model which is used in AES modelling for TF-IDF and TextRank summary data.

3.3 Clustering Model Analysis and Final Result

The AES modelling was re-examined with three forms of data to analyse the effect of the summary model results. The three forms of data are original essay data, TF-IDF summary results, and TextRank summary results which have been combined with the readability index assessment. Table 7 displays the results in the form of a silhouette score as a form of analysis of clustering results.

Table 7 Clustering results with AES model

Data	Silhoutte score_1	Silhoutte score_2	Silhouette score_3	Average
Original form	**0.41587745**	**0.727286113**	**0.69264041**	**0.611934657**
TF-IDF Summary	0.529353676	0.538175068	0.68056125	0.582696665
TextRank Summary	0.485848456	0.720769021	0.597122163	0.601246546

In 3 evaluations, the model with the original essay data had a better average silhouette score, which was 0.611934657. The best silhouette score in the second evaluation was obtained by AES with the original essay data, which was 0.727286113, which only had a difference of 0.006517092 with AES TextRank summary data. This value is also the highest value of the overall value obtained in the evaluation process. The average value of the AES modelling of the original essay data with 2 forms of summary data shows that the difference is not too large. The difference between the original silhouette score scores and the TF-IDF summary is 0.029237992, while the TextRank summary is 0.010688111. This indicates that there is a potential for increasing the silhouette score with the use of summary data in the AES of this study. However, this study shows that AES with original essay data is still better than the use of summary data on AES.

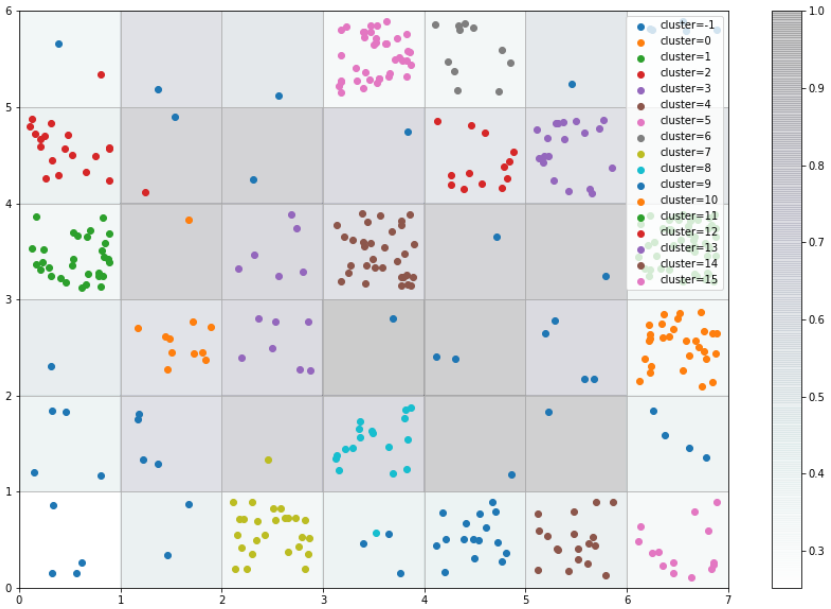

Figure 6 Mapping result cluster

AES with the original essay data resulted in 15 clusters with some noise data. Data points that are considered as noise are defined with a value of -1 and are coloured dark blue scattered

over each square of the mapping area shown in Figure 6. The number of noise or outliers is 51 data points.

Based on Table 8, each cluster can be grouped into a new group according to the value of each index. The FRE index shows a text will be more difficult to understand in the index range 0-30, while the higher the index of the other four indexes, the more difficult it is to understand. However, in this case study, it is hoped that the essay can achieve this difficulty because it is written by an academic. Even so, the essay is still easy to understand by assessors at a balanced level. In the index value analysis process, Cluster 0 meets the criteria for the expected index value.

Table 8 Average readability index and silhouette score for each cluster

Klaster	FRE	*Flesch-kincaid*	*Gunning-fog*	ARI	*Coleman-liau*	*Silhoutte score*
0	**49.33830074**	**12.50532918**	**15.252031**	**13.33470359**	**10.29504934**	**0.842763544**
1	69.69103499	7.673007957	10.5454577	6.928334633	7.021425566	0.519504365
2	60.90816016	10.07702706	13.03145774	10.02843946	8.17411041	1
3	63.06621707	8.88770867	11.72831192	8.579058153	8.446742921	0.830059009
4	52.07282659	11.25095384	14.07418173	11.75872797	10.44333469	1
5	70.45550148	7.542167779	10.35637529	6.770547024	6.872660616	0.924585219
6	67.55429575	8.384905958	11.20351271	7.938482447	7.343830312	1
7	63.69828931	9.224285371	11.96875825	8.98269264	7.939846632	0.928571429
8	67.73208409	8.010977695	10.78920321	7.480159252	7.562175563	0.829799012
9	71.48798361	7.695276118	10.51354539	7.01439045	6.497260832	0.863636364
10	59.63849761	9.958464532	12.84870288	9.834662929	8.608704674	1
11	68.52911549	8.122429263	11.13605977	7.603002854	7.258765431	0.869747215
12	60.81761659	9.779094888	12.66051656	9.649143891	8.437237857	1
13	54.78947196	10.67311924	13.4345775	10.82276327	9.799276506	0.530678751
14	60.95488981	9.621698519	12.42881147	9.648988529	8.761671135	1

In addition to using a readability index approach, a content analysis approach is also carried out by determining the common phrases mentioned in the essay collection of each cluster. The phrase is formed in the definition of n-gram = 3. This is done to meet criteria related to content keywords, namely passion, vision, and action. The phrases "enablers of passion" and "passion for teaching" are common phrases that appear in almost every cluster related to passion. The phrases "to be remembered", "to remember me", "to become" appear frequently in each cluster and are associated with "student/s" related to vision. "I need to", "I have to", "make a difference" are phrases that have had many occurrences and are related to the content approach with the keyword action. However, the phrases already mentioned are too general to define as an approach to explaining the content of each keyword. Clusters 0, 3, 4, 5, and 14 mention the word "family" which can be used as a reference to define passion. Clusters 0, 2, and 14 mention the phrase "become a better" which defines more fully about vision. In addition, Clusters 2 and 4 also mention activities to be carried out with the phrases "want to achieve" and "achieving their dreams" which explain in more detail about the action. The resulting analysis shows that Clusters 0, 2, 3, 4, and 14 are potential documents that can be defined well based on the assessment of the content approach.

Based on the analytical approach to assessing essays based on the two assessment criteria, it was found that Clusters 0, 2, 3, 4, and 14 have good potential to be defined as the best collection of documents. However, Cluster 0 has a superior point, which both approaches define well. Thus, Cluster 0 is the best collection of documents in this research case study.

4. CONCLUSIONS

Based on the research that has been done, it can be concluded that ATS does not have a good effect on AES in the form of clustering in increasing the silhouette score value for TEACH

ON essays, where the best AES modelling results are obtained by using the original essay data which produces an average silhouette score of 0.612. However, it can be seen that ATS has the potential to increase the silhouette score in AES in the form of clustering, seeing that the difference in scores is not too large. In the results of the analysis of grouping documents with the original essay data, Cluster 0 is the best cluster based on an analytical approach using a readability index and extraction of common phrases with n-gram = 3 which is carried out to meet the criteria for assessing cohesion of ideas and completeness.

Based on the case studies raised in this study, it can be said that larger and larger amounts of data with good measurable quality are needed to build a better AES model. In addition, AES modelling requires a defined score so that the learning process becomes more focused. The definition of the score has an effect on forming good data as an input for the assessment model for a good learning process. In the process of analysing the influence of ATS in AES as is done in research, it is necessary to summarize the targets of the essay, so that there are features that can be studied to get the best summary results. In addition, with the existence of a summary target, the summary model implemented is more diverse.

REFERENCES

[1] Ke, Z. & Ng, V., 2019, Automated essay scoring: A survey of the state of the art, *IJCAI International Joint Conference on Artificial Intelligence*, 2019-August, 6300–6308.

[2] Kumar, V. & Boulanger, D., 2020, Explainable Automated Essay Scoring: Deep Learning Really Has Pedagogical Value, *Frontiers in Education*, 5, October, 1–22.

[3] Zhou, S., Jeong, H. & Green, P.A., 2017, How consistent are the best-known readability equations in estimating the readability of design standards?, *IEEE Transactions on Professional Communication*, 60, 1, 97–111.

[4] Beinborn, L., Zesch, T. & Gurevych, I., 2014, Readability for foreign language learning, *ITL - International Journal of Applied Linguistics*, 165, 2, 136–162.

[5] Zhao, S., Zhang, Y., Xiong, X., Botelho, A. & Heffernan, N., 2017, A memory-Augmented neural model for automated grading, *L@S 2017 - Proceedings of the 4th (2017) ACM Conference on Learning at Scale*, 189–192.

[6] West-Smith, P., Butler, S. & Mayfield, E., 2018, Trustworthy Automated Essay Scoring without Explicit Construct Validity, 95–102. *www.aaai.org*.

[7] Nemati, M. & Azizi, M., 2013, Readability index of essays as an alternative to the scoring procedure in L2 academic writing, 4, Winter, 2–10.

[8] Le, Q. & Mikolov, T., 2014, Distributed representations of sentences and documents, *31st International Conference on Machine Learning, ICML 2014*, 4, 2931–2939.

[9] Prabowo, D.A., Fhadli, M., Najib, M.A., Fauzi, H.A. & Cholissodin, I., 2016, TF-IDF-Enhanced Genetic Algorithm Untuk Extractive Automatic Text Summarization, *Jurnal Teknologi Informasi dan Ilmu Komputer*, 3, 3, 208.

[10] Mihalcea, R. & Tarau, P., 2004, TextRank : Bringing Order into Text, *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 404–411. *https://www.aclweb.org/anthology/W04-3252*.

[11] Astudillo, C.A. & Oommen, B.J., 2014, Topology-oriented self-organizing maps: A survey, *Pattern Analysis and Applications*, 17, 2, 223–248.

[12] Asan, U., Soyer, A. & Serdarasan, S., 2012, *Computational Intelligence Systems in Industrial Engineering*, C. Kahraman, ed., Atlantis Press, Paris. *http://www.springerlink.com/index/10.2991/978-94-91216-77-0*,.

[13] Campello, R.J.G.B., Moulavi, D. & Sander, J., 2013, Density-based clustering based on hierarchical density estimates, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7819 LNAI, PART 2, 160–172.