Received: January 1, 2020. Revised: January 27, 2020.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Indonesian Short Essay Scoring Using Transfer Learning Dependency Tree LSTM

Agung Wiratmo[1] **Chastine Fatichah**[1]*

[1] *Department of Informatics, Institut Teknologi Sepuluh Nopember, Indonesia*
* Corresponding author's Email: chastine@if.its.ac.id

Abstract: Distributed representation of a sentences cannot only be seen with the sequence but also with dependency. In this paper, we proposed an answer assessment model that considers with dependency relational each word. The dependency relational is obtained by universal dependency modelling from CoNLL format data. The dependency relational is used in Long Short-Term Memory (LSTM) architecture by modified the hidden state, which is called dependency tree LSTM. The proposed method has an improvement on QWK and accuracy in 2.38% and 2.05%, respectively, that compared with the LSTM state of the art in the English short essay. Furthermore, the proposed method with an Indonesian short essay shows the evaluation of QWK and accuracy of 68.07% and 82.51%, respectively.

Keywords: Short essay assessment, Transfer learning, Dependency relational, Dependency tree LSTM.

1. Introduction

Assessment of learning outcomes, which is conducted by a teacher is aimed to evaluate, learning monitor, and improve learning outcomes. A Short essay is one of the evaluation methods for this assessment. Assessment with short essays can cause several problems as time consumption, greater subjectivity, and change the assessment criterion.

There are several previous kinds of research in the assessment of short essays that used conventional [1, 2] and deep learning approaches [3–6]. The conventional approaches are used feature from several methods such as Term Frequency-Inverse Document Frequency-Document Frequency (TF-IDF-DF) [1], etc.

The deep learning approaches make use of the feature that is obtained from distribution representation. Many researchers are used neural networks in this study. The deep learning architecture that mainly used is a Convolutional Neural Network (CNN) [6], Recursive Neural Network (RNN) [7, 8] and the combination of both [3–5]. The most commonly RNN used is Long Short-Term Memory (LSTM). Not only used that architecture several kinds of research but also used modification on attention mechanism, hierarchical structure, and coherence feature, etc.

Researches are used a supervised approach that has some tasks such as regression [3, 7] and classification [3, 4, 6]. The goal of the regression task is to predict the score. While the goal of the classification task is to classify into several classes.

Mueller, et al. [7] demonstrated Manhattan LSTM for scoring the similarity of sentences that reflect semantics. This approach compares two sentences in each network that have sequence word relational distribution of these sentences[7]. This approach performs a regression task that predicts the similarity of sentences with other sentences.

Dong, et al. [6] explore hierarchical CNN which consists of two-layer. The lower layer is a depiction the sentence representation. The upper layer is a depiction structure based on sentence representation. The first step of this approach computes the important feature from each word vector in the sentences. The important local feature is obtained using computation based on the window [6]. After getting the important

You ⟶ may ⟶ not ⟶ sublicense ⟶ the ⟶ Work ⟶ .

Figure. 1 Example of sequence relation on the sentence

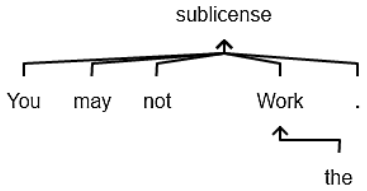

Figure. 2 Example of dependency relation on the sentence

local feature, this feature is used on a higher layer that is a sentence layer that concatenates the importance of local features of the word in these sentences. This layer used the same schema of computed to obtain the important local feature. Then, the highest layer classifies the score using fully connected that is obtained from the important local feature.

Bui, et al [8] propose a deep neural network approach for the classification of antonyms and synonyms using co-occurrence context and word structure. Both co-occurrence context and word structure, each word in sentences are represented by a vector of word embedding and part of speech (POS) representation. The feature on co-occurrent context consists of word representations of synonym and antonym in the sentences and the feature of word representation of all word in sentences obtained by bidirectional LSTM.

The feature on word structure is developed by vector construction function including a dot-product similarity value between two words representation transformed by the sigmoid function, a Lexicographers Mutual Information score, and a word-structure pattern encoded value [8]. The feature concatenation of co-occurrence context and word structure used to classify antonyms and synonyms.

Riordon, et al. [3] investigate several basic deep learning approaches that combine CNN and Long Short-Term Memory (LSTM) method with attention and mean over time mechanism. LSTM is the development of RNN. In this approach, computation with CNN obtains the important local feature of each word but this step is optional. The feature obtained from CNN or directly from the word vector is used for classifying or predicting the score. This approach also used the mean over time that is the vector value of the hidden state on the LSTM is the average of all previous hidden states.

Liang, et al. [4] proposed a Siamese Bidirectional Long Short-Term Memory (SBLSTMA). This method is counting the similarity of the essay and the sample and explore a self-feature mechanism. In this approach, computation with the CNN that did before the LSTM obtains the important local feature of each word, but this step is optional. CNN is done to handle sentences that are too long. The self-feature mechanism consists of feature each network, cross-feature, and inner-feature. Cross-feature is computed from the cosine similarity of features between each network. Inner-feature is computed from the cosine similarity of feature in answer and the previous answer. The features that are obtained from concatenation these three features are used in the score classification process.

Distribution representation on RNN, that usually used is sequence distribution in unidirectional [3, 7] or bidirectional [4, 8]. Other than that, the representation can use dependency distribution. The dependency distribution [9] is composed of the representation of the sentence into its constituent words to getting syntactic structure. An example of the sentence “You may not sublicense the Work.”, the distribution relational can be shown in sequence like Fig. 1. and in dependency like Fig. 2.

Recent Studies which is used deep learning approach the most used sequence distribution and not consider the feature POS [3, 4, 6, 7]. Whereas considering the use of the POS feature [8] are only used as additional features not to determine the distribution representation. We assume that the distribution of sequences is inadequate to capture the semantics of sentences. This is due to the lack of the sequence distribution ability to distinguish the meaning of different sentences in the syntactic sequence or structure as an example “the man drinks coffee” vs “coffee drinks the man”.

In this paper, we used the dependency relation to get the feature of syntactic structure in the sentences using POS. That is not usually used in the previous study. On the other hand, this paper mainly used in the Indonesian short essay. But because of the lack of short essay data is challenged. Because of that, we used transfer learning from the English short essay

id	form	lemma	upostag	xpostag	feats	head	deprel	deps	misc
1	Pinisi	_	PROPN	_	_	3	nsubj	_	_
2	sebenarnya	_	ADV	_	_	3	advmod	_	_
3	merupakan	_	VERB	_	_	0	root	_	_
4	nama	_	NOUN	_	_	3	obj	_	_
5	layar	_	NOUN	_	_	4	compound	_	_
6	.	_	PUNCT	_	_	3	punct	_	_

Figure.3 Indonesian corpus in CoNLL format syntax

that commonly used in the previous study. In this data, certainly we treatment like Indonesian short answer.

So, in this paper, we proposed an answer assessment model that considers with dependency relational each word to get syntactic structure of each answer which is called dependency tree LSTM. Due to Indonesian short essays has a small amount of data, transfer learning is used in this paper. Transfer learning is a technique to share knowledge from data source to data target. The data source is a data from an English short essay that getting from Kaggle. The data target is an Indonesian short essay.

2. Related works

2.1 Data augmentation

The availability of good and balance data set is the key to build an optimal model but not all correspond to reality [10]. There are at least two methods to handle this problem that is oversampling and under sampling. Oversampling is a method to create more data from the lower data frequency [11]. Under sampling is the opposite method to over sampling that to omitted data in the higher data frequency [12].

Augmentation is one of the methods on oversampling that appends new data with synthetic data. Several simple ways to create synthetic data with augmentation method are substitute, insert n word with a synonym of the word, swap the $n$word position, and delete the n word [13].

2.2 Dependency parsing

The grammatical structure can establish with constituency structures and dependency structures [14]. In constituency structures, the grammatical structure of sentences is obtained by grouping word into nested constituencies[15]. Another case with dependency structure, the grammatical structure is contracted by seeking each word relational[15]. Dependency parsing is an analysis that based on structural dependency syntax in the sentences [16]. The dependency parsing output is a parse of sentence that based on POS tag and shown by the tree diagram. There are two steps in dependency parsing process that is learning and parsing process.

The learning process is a process to create a model from training data, which contains dependency relational. The training data is a corpus which contains CoNLL format that shown in Fig. 2. That model is obtained by the transition-based neural method that the input data are words x^w, the POS tag x^t, arc labels x^l, word weighted W^w, POS tag weighted W^t, label weighted W^l and bias b. Whereas, h obtains a hidden state and p obtain a dependency arch that is an output layer. The equation of transition-based neural is shown in Eqs. (1) and (2).

$$h = \left(W^w x^w + W^t x^t + W^l x^l + b\right)^3 \quad (1)$$

$$p = softmax(Wh) \quad (2)$$

2.3 Dependency tree-LSTM

LSTM is a deep learning algorithm that modified from Recurrent Neural Network (RNN) [17]. Dependency Tree-LSTM is one of this modification in the hidden state. In LSTM, the hidden state is weighted by sequence and has been relational only with a previous hidden state. Whereas in dependency tree-LSTM, hidden state has relation with leaf node hidden state $\Sigma_{a \epsilon C_p} h_a$ whereas C_p is a leaf node from the parent and h_a is a hidden state of parent node.

$$i_t = \sigma(W^x . x_t + W^h . \Sigma_{a \epsilon C_p} h_a + b) \quad (3)$$

$$o_t = \sigma(W^x . x_t + W^h . \Sigma_{a \epsilon C_p} h_a + b) \quad (4)$$

$$\widetilde{c_t} = tanh(W^x . x_t + W^h . \Sigma_{a \epsilon C_p} h_a + b) \quad (5)$$

$$f_{ta} = \sigma(W^x . x_t + W^h . h_a + b) \quad (6)$$

$$c_t = i_t \odot \widetilde{c_t} + \Sigma_{a \epsilon C_p} f_{ta} \odot c_a \quad (7)$$

$$h_t = o_t \odot \tanh(c_t) \quad (8)$$

Where i_t, o_t, c_t, f_t, h_t in Eq. (3)-(8) is an input gate, output gate, memory cell, forget gate, and hidden state, respectively that has three input data that is input vector x_t, hidden vector h_a, and bias b and also has two weighting that is input weight W^x and hidden weight W^h. Moreover σ, $tanh$, and $\odot$ respectively donates a sigmoid, hyperbolic tangent activation function and element-wise multiplication.

2.4 Transfer learning

Transfer learning is aimed to transfer knowledge from one to another [18]. There are two transfer learning methods that are traditional learning and transfer learning. Traditional learning is a learning method to solve the specific problem. Transfer learning is a learning method to solve the problem that is used for other problems. There are several types of transfer learning that is domain adaption, cross-lingual learning, multitask learning, and sequential transfer learning [18].

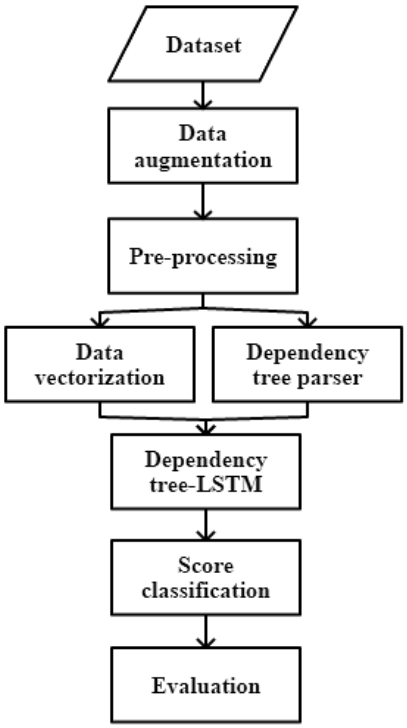

Figure. 4 The framework of the proposed method

Domain adaption can be implemented on target with either or no minimal label data. Cross-language leaning can be implemented in a different language. Multitask learning can be implemented on leverage data available in different domains. Sequence transfer learning improves the transferring knowledge with a sequence of stride where source and target are not necessarily the same.

2.5 Performance evaluation

Performance evaluation on this paper is accuracy (*Acc*) and quadratic weighted kappa (QWK). Accuracy is compared count of correct prediction with count of all data. QWK is an agreement measurement between label value i with prediction value j [19]. QWK is weighted the quadratic weighted matrix $W_{i,j}$ that shown in Eq. (10).

$$W_{i,j} = \frac{(i-j)^2}{(N-1)^2} \tag{10}$$

$$QWK = 1 - \frac{\Sigma W_{i,j} O_{i,j}}{\Sigma W_{i,j} E_{i,j}} \tag{11}$$

N is a count of unique on label value. QWK is obtained from $W_{i,j}$, $O_{i,j}$, and $E_{i,j}$. $O_{i,j}$ is the sum of value that must label i and prediction j. $E_{i,j}$ is a division of $O_{i,j}$ by total frequency label i in all data that shown in Eq. (11).

Table 1. Example of ASAP-SAS data set

Student answer	Score
You would need many more pieces of information to replicate the experiment. You would need the type of samples to begin with in the procedure. You would also need to know the amount of vinegar used in each container.	3
Some additional information that I would need is the amount of vinegar they poured.	1

Table 2. Example of Indonesian short answer data set

Student answer	Score
sutradara : mengatur jalannya pementasan ; alur : jalan cerita; panggung : tempat pementasan drama; amanat : pesan yang terkandung dalam drama	5
bersifat fakta; terdiri dari jawaban 5w+1h; benar-benar terjadi	2

3. Research method

In this paper, the framework of the proposed method is showed in Fig. 4 that contains several steps which will be discussed below.

3.1 Dataset description

There are two types of dataset used throughout this paper that are source and target data set. Source data set is an Automatic Student Assessment Prize Short Answer Scoring (ASAP-SAS) data set from Kaggle that shown in Table 1. Target data set is Indonesian learning outcome that is formed short answer that shown in Table 2. To handle small data set in target data, source data set is used.

The amount of data in source data set is 17207 and target data set is 1436. Responses of source data set are scored by two human annotators on a scale from 0 to 2 or 0 to 3 depending on the prompt. Responses of target data set are scored by human annotators on a scare from 2 to 3 on all the prompt.

3.2 Data augmentation

Data augmentation is only done in training process in data target that is to handle the unbalance data set and prevent misclassification on lower frequency data set. That process is done in 4 ways, namely insert, substitute n word by this synonym, position n word swapping, and delete n word [13].

3.3 Pre-processing

Pre-processing process is done in three steps, namely remove the special character, change the number to word, and perform tokenization that illustrated in Table 3. Pre-processing in this paper is

Table 3. Illustration of pre-processing data

Original data	Pre-processing data
rumus : jumlah kata yang dibaca/ jumlah waktu yang dibutuhkan x 60 = 240 / 120 x 60 = 120 kpm	['rumus', 'jumlah', 'kata', 'yang', 'dibaca', 'jumlah', 'waktu', 'yang', 'dibutuhkan', 'x', 'enam', 'puluh', 'dua', 'ratus', 'empat', 'puluh', 'seratus', 'dua', 'puluh', 'x', 'enam', 'puluh', 'seratus', 'dua', 'puluh', 'kpm']

not done case folding and stop word because that sometime important.

3.4 Dependency tree parser

To create a dependency tree parser in the tree diagram, this initialization step is made model with transition-based neural network, defined [ROOT] in stack, and defined empty [] in relation. The data is used for this model is English and Indonesian corpus in CoNLL format. This data contains word, POST tag, and arch label. Then the concatenation of three data is used in the first layer in this network. Then the next layer is count hidden layer that uses cubic activation. The output layer is a dependency arch. Dependency arch is used arc-standard system [20] that is left arch, right arch and shift.

The first step in a dependency parser is a word list that made from tokenization of sentence. The second step is to check relation each token in the stack using the model that previously made. The relation is left arch, right arch or shift. Left arch is a relation dependency from stack to be the previous stack. Right arch is a relation dependency from stack to be next stack. However, if it has not relation that call stack.

Third step is changing the stack that depends on the relation it has. If it has left arch, the previous last stack is deleted. If it has left arch, the last stack is deleted. If it has shift, first token in the word list is appended in the stack. That process is repeated until stack has [ROOT] and empty in the word list.

Table 4. Training hyper-parameter

Layer	Parameter	
	Name	Value
Embedding layer	Pretrained embedding	GloVe 250 dimensional [3]
LSTM layer	Layer	1 [4]
	Hidden units	256
	Dropout	0.5
	Epochs	50
	Batch size	32
	Learning rate	10^{-2}
	Optimisation	Adaptive Gradient [21]
	Weight decay	10^{-2}

3.5 Modelling

3.5.1. Modelling on source data

In this process, the input data is a vector of each word in each sentence in each prompt in source data set and the dependency parser of this sentence. A data vector is established using GloVe 250 dimensions. Dependency parser is a linked list of each word in the sentence.

The next step is modelling in source data set using dependency tree-LSTM that shown in Eqs. (3)-(8) and the hyperparameter is shown in Table 4. The initialization on weighted of these equations is random but after each batch sizes the weight is updated. Updating weight is influenced by several parameters one of which is the loss method. In this paper is used cross entropy for multiple categories. After processing in tree-LSTM, the output of this process is a hidden state h_t. After getting value of hidden state h_t , the next process is scoring classification. This classification uses SoftMax activation. The output of this process is getting the score.

3.5.2. Transfer learning in target data set

In this process, the input data is a vector of each word in each sentence in each prompt in target data set and the dependency parser of this sentence. Vector is established using GloVe 250 dimensions. Dependency parser is a linked list of each word in sentence.

Different from the above process, the initialization on weighted of the process is used the source dataset weight but after each batch size, the weight is updated. Updating weight is influenced by several parameters one of which is the loss method. In this paper is used cross entropy for multiple categories. After processing in tree-LSTM, the output for this process is a hidden state h_t. After getting value of hidden state h_t, the next process is scoring classification. This classification uses SoftMax activation. The output from this process is getting the score.

4. Result and discussion

This paper was implemented on python using pytorch library, GloVe, and java using Stanford CoreNLP. The specification of the machine on which was run is Intel (R) Xeon (R) CPU @ 2.30 GHz, GPU 1 x Tesla K80 2495, and RAM 12.6 GB.

Table 5. Statistics of source and target dataset

#	Source dataset			Target dataset		
	Train	**Test**	**Score range**	**Train**	**Test**	**Score range**
1	1672	557	0-3	105	30	2-5
2	1278	426	0-3	107	28	2-5
3	1891	406	0-2	112	23	2-5
4	1738	295	0-2	102	33	2-5
5	1795	598	0-3	109	26	2-5
6	1797	599	0-3	130	22	2-5
7	1799	599	0-2	128	24	2-5
8	1799	599	0-2	114	38	2-5
9	1798	599	0-2	120	32	2-5
10	1640	546	0-2	121	31	2-5

Table 6. Comparison of deep learning method on source

Network	**Performance (%)**	
	Mean QWK	**Mean *Acc***
LSTM [3]	60.47	66.38
SBLSTMA [4]	9.17	52.20
Proposed method	62.85	68.43

Dataset used in this paper is source and target dataset that already explained in section 3.1. Each dataset contains ten prompts. Each prompt contains different range of score that described in Table 5.

In this paper, to pre-train word embedding, we were used Stanfold GloVe 250-dimensional embedding [3]. This paper is used 5-fold cross validation [3]. To pre-train model transition-base neural network on dependency parser, universal dependency data set with ConLL format was used.

The Universal dependency in English language and Indonesia language is used UD_English-ParTUT and Indonesian UD, respectively. The Evaluation metrics of this model is Unlabeled Attachment Score (UAS) and Labeled Attachment Score (LAS). Indonesian model is getting score 72.78% and 64.19% of UAS and LAS, respectively. English model is getting score 74.78% and 69.09% of UAS and LAS, respectively.

After getting the model transition-base neural network, this model is used to getting the dependency parser of source and target data. This dependency parser and word embedding is used in proposed method. Scoring model is train on each score range in LSTM [3], SBLSTMA [4], and the proposed method. The hyper-parameter LSTM that the previous method is used in this paper is RMSProp optimizer with the value of learning rate, weighted decay, batch size, and dropout are10^{-3},10^{-3}, 32, and 0.5, respectively and implement mean over time.

The hyper-parameter SBLSTMA that the previous method is used in this paper is Adagrad optimizer with the value of learning rate, weighted decay, batch size, and dropout are10^{-2},10^{-3}, 32, and 0.75, respectively. Using source data set, the previous method is compered by the proposed method that the result shown in Table 6. In this paper, evaluations are obtained from models that have the best QWK evaluation in training.

Table 7. Comparison of deep learning on each prompt in source data set

Prompt	**Network Performance (%)**					
	LSTM[3]		**SBLSTMA**[4]		**Proposed method**	
	QWK	***Acc***	**QWK**	***Acc***	**QWK**	***Acc***
1	67.16	52.93	18.58	33.57	69.36	53.79
2	64.61	52.58	15.88	34.41	67.12	52.58
3	24.20	52.51	0.85	55.42	36.89	60.30
4	58.48	69.69	7.37	50.31	58.30	71.32
5	74.63	82.58	1.46	77.39	74.76	83.21
6	74.88	85.38	0	83.14	75.00	87.18
7	52.31	64.14	2.49	49.55	52.45	65.41
8	48.88	59.33	23.51	51.02	52.62	63.47
9	71.89	69.82	12.34	40.40	73.50	70.35
10	67.71	74.80	9.19	46.81	68.50	76.70
Mean						
	60.47	66.38	9.17	52.20	62.85	68.43
Standard deviation						
	15.54	12.18	8.23	16.48	12.49	11.62

In Table 6, the proposed method has the best result on QWK and accuracy, 62.85% and 68.43%, respectively. The evaluation on LSTM has 60.47% and 66.38% in QWK and accuracy, respectively. While the SBLSTMA evaluation has 9.17% and 52.20% in QWK and accuracy, respectively. In the third, methods have a small evaluation of QWK and accuracy because cannot get out overfitting, although, in the method of prevention with 5-fold cross validation, weighted decay and decay are used. The detail evaluation in each prompt is shown in Table 7.

In Table 7, the detail evaluation on LSTM network has standard deviation on QWK 15.54% that show the highest variant of QWK. While, the smallest variant that shown in standard deviation value on QWK is on SBLSTMA Network. That is because in this network weighted the vector of student answer and the vector of different of student answer and the true answer and weighting the different of the current and previous feature of the student answer. Furthermore, the parameter of that network in previous research is used on essay data set but in this paper, we adjust only on batch size. That is caused the under fitting on SBLSTMA network.

Whereas the standard deviation of detail evaluation on accuracy on SBLSTMA has the highest value that inversely proportional in QWK. That shown the predicted score on SBLSTMA has high difference misclassification. Furthermore, the

Table 8. Evaluation model on data target without transfer learning on QWK and accuracy

Data	Performance (%)			
	No Transfer Learning		With Transfer Learning	
	Mean QWK	Mean *Acc*	Mean QWK	Mean *Acc*
No augment	48.26	81.99	64.58	84.74
Augment				
Substitute	55.72	79.72	63.40	81.98
Swap	58.51	80.29	66.12	82.80
Delete	65.26	82.50	67.94	80.56
Insert	60.06	78.67	68.07	82.51

Table 9. Standard deviation model on data target transfer learning.

Data	Performance (%)			
	No Transfer Learning		With Transfer Learning	
	σ QWK	σ *Acc*	σ QWK	σ *Acc*
No augment	39.56	17.93	35.28	17.92
Augment				
Substitute	35.97	18.10	28.51	14.20
Swap	33.24	16.34	32.11	18.48
Delete	29.56	16.33	31.66	17.45
Insert	29.22	16.80	21.14	12.23

smallest standard deviation on accuracy is shown on proposed method.

After getting the weight in the source model, it is used as weight initialization on the target model. Before that the process must be carried out. The pre-train a process is creating the data augmented on data training in target data. The network is used in target data is the best network in source data that is proposed method.

To create a model on target data, two scenarios are with or without transfer learning and augmentation data. That evaluation means of QWK and accuracy on target data is shown in Table 8 and standard deviation QWK and accuracy on target data is shown in Table 9. The augmentation data is created with substitute, swap, delete, or insert new data with a synonym.

In Table 8, it shown the mean evaluation on QWK and accuracy that shown model with transfer learning getting the highest QWK and accuracy on 68.07% and 84.74%, respectively. However, the standard deviation on this evaluation shown on Table 9 that shown the best value of standard deviation QWK and accuracy is on data with augmented using insert. It be caused the important word maintained although the synonym word is inserted.

Creating model with transfer learning show improvement then without it. The transfer learning model that used in the target model is the range score 1-3 or four categories and the target model in the range score 2-5 or four categories. That model into the source is used because has a same count of category. The transfer learning causes an increase evaluation.

In Table 9, it has shown the standard deviation in the target model. It has shown the best standard deviation is a model with transfer learning and used the augmented data using an insert synonym. That value is shown differentiation of the specific evaluation is small. That happened is caused the important of data and that position maintained.

5. Conclusion

A new strategy for Indonesian short essay scoring using transfer learning dependency tree LSTM was presented on this paper. Based on the experimental result, the proposed method has the best model evaluation on source and target data. The architecture in LSTM for scoring essay can consider not only sequence but also dependency. In the source model compared with LSTM that is the state of the art, this proposed method has better QWK and accuracy on 2.38% and 2.05%, respectively. Furthermore, if compared with SBLSTMA that is the state of the art, this proposed method has better QWK and accuracy result on 53.68% and 16.23%, respectively.

In the target model, we only used the best network on the source model that is the proposed method. In this target model, used the augmentation data with transfer learning has a better result. The best evaluation in the target model is used augmentation data with inserted the synonym in QWK evaluation. That evaluation shown in the model with transfer learning and augmented data with an inserted synonym has mean QWK and accuracy in 68.07% and 82.51%, respectively. Beside that also has standard deviation on QWK and accuracy in 21.14% and 12.23%, respectively. For future work, this research must consider not only one feature such as the syntactic structure but must consider several features in the rubric.

References

[1] A. R. Lahitani, A. E. Permanasari, and N. A. Setiawan, "Cosine similarity to determine similarity measure: Study case in online essay assessment" , In: *Proc. of 2016 4th International Conference on Cyber and IT Service Management*, pp. 1–6, 2016.

[2] F. S. Pribadi, A. E. Permanasari, and T. B. Adji, "Short answer scoring system using automatic reference answer generation and geometric

average normalized-longest common subsequence (GAN-LCS)", *Education and Information Technologies*, Vol. 23, No. 6, pp. 2855–2866, 2018 .

[3] B. Riordan, A. Horbach, A. Cahill, T. Zesch, and C. M. Lee, "Investigating neural architectures for short answer scoring" , In: *Proc. of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 159–168, 2018.

[4] G. Liang, B. W. On, D. Jeong, H. C. Kim, and G. S. Choi, "Automated essay scoring: A siamese bidirectional LSTM neural network architecture", *Symmetry*, Vol. 10, No. 12, pp. 1–16, 2018 .

[5] F. Dong, Y. Zhang, and J. Yang, "Attention-based Recurrent Convolutional Neural Network for Automatic Essay Scoring" , In: *Proc. of the 21st Conference on Computational Natural Language Learning*, pp. 153–162, 2017.

[6] F. Dong and Y. Zhang, "Automatic Features for Essay Scoring-An Empirical Study", In: *Proc. of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1072–1077, 2016.

[7] J. Mueller, A. Thyagarajan, and Jonas Mueller, "Siamese Recurrent Architectures for Learning Sentence Similarity", In: *Proc. of the Thirtieth AAAI Conference on Artificial Intelligence*, No. 2012, pp. 1386–1393, 2014.

[8] V.-T. Bui, P.-T. Nguyen, V.-L. Pham, and T.-Q. Ngo, "A Neural Network Model for Efficient Antonymy-Synonymy Classification by Exploiting Co-occurrence Contexts and Word-Structure Patterns", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 1, pp. 156–166, 2020 .

[9] K. S. Tai, R. Socher, and C. D. Manning, "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks", In: *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp. 1556–1566, 2015.

[10] X. Zhang, J. Zhao, and Y. Lecun, "Character-level Convolutional Networks for Text Classification", In: *Advances in Neural Information Processing Systems*, pp. 649–657, 2015.

[11] Z. Zheng, Y. Cai, and Y. Li, "Oversampling Method for Imbalanced Classification", *Computing and Informatics*, Vol. 34, No. 5, pp. 1017–1037, 2016 .

[12] B. Santoso, H. Wijayanto, K. Notodiputro, and B. Sartono, "Synthetic over sampling methods for handling class imbalanced problems: a review", *IOP Conference Series: Earth and Environmental Science*, Vol. 58, pp. 1–8, 2017.

[13] J. W. Wei and K. Zou, "EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks", In: *Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 6381–6387, 2019.

[14] D. Klein and C. D. Manning, "Corpus-based induction of syntactic structure: Models of constituency and dependency" , In: *Proc. of the 42nd Annual Meeting on Association for Computational Linguistics*, Vol. 5, p. 478, 2004.

[15] J. Zhou and H. Zhao, "Head-Driven Phrase Structure Grammar Parsing on Penn Treebank", In: *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2396–2408, 2019.

[16] D. Chen and C. Manning, "A fast and accurate dependency parser using neural networks", In: *Proc. of the 2014 conference on empirical methods in natural language processing*, pp. 740–750, 2014.

[17] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, 1997 .

[18] S. J. Pan and Q. Yang, "A Survey on Transfer Learning", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 22, No. 10, pp. 1345–1359, 2009 .

[19] J. Cohen, "A coefficient of agreement for nominal scales", *Educational and psychological measurement*, Vol. 20, No. 1, pp. 37–46, 1960 .

[20] J. Nivre, J. Hall, and J. Nilsson, "MaltParser : A Data-Driven Parser-Generator for Dependency Parsing" , *LREC*, No. 6, pp. 2216–2219, 2006.

[21] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization", *Journal of Machine Learning Research*, Vol. 12, pp. 2121–2159, 2011 .