Hybrid Deep Learning CNN-Bidirectional LSTM and Manhattan Distance for Japanese Automated Short Answer Grading

Use case in Japanese Language Studies

Anak Agung Putri Ratna*
Department of Electrical Engineering, Universitas Indonesia, Depok, Indonesia
ratna@eng.ui.ac.id

Prima Dewi Purnamasari
Department of Electrical Engineering, Universitas Indonesia, Depok, Indonesia
prima.dp@ui.ac.id

Nadhifa Khalisha Anandra
Department of Electrical Engineering, Universitas Indonesia, Depok, Indonesia
nadhifa.khalisha@ui.ac.id

Dyah Lalita Luhurkinanti
Department of Electrical Engineering, Universitas Indonesia, Depok, Indonesia
dyah.lalita11@ui.ac.id

ABSTRACT

This paper discusses the development of an Automatic Essay Grading System (SIMPLE-O) designed using hybrid CNN and Bidirectional LSTM and Manhattan Distance for Japanese language course essay grading. The most stable and best model is trained using hyperparameters with kernel sizes of 5, filters or CNN outputs of 64, a pool size of 4, Bidirectional LSTM units of 50, and a batch size of 64. The deep learning model is trained using the Adam optimizer with a learning rate of 0.001, an epoch of 25, and using an L1 regularization of 0.01. The average error obtained is 29%.

CCS CONCEPTS

• **Applied Computing**; • **Education**; • **E-learning**;

KEYWORDS

CNN, BiLSTM, Automated Short Answer Grading

ACM Reference Format:
Anak Agung Putri Ratna, Prima Dewi Purnamasari, Nadhifa Khalisha Anandra, and Dyah Lalita Luhurkinanti. 2022. Hybrid Deep Learning CNN-Bidirectional LSTM and Manhattan Distance for Japanese Automated Short Answer Grading: Use case in Japanese Language Studies. In *2022 the 8th International Conference on Communication and Information Processing (ICCIP 2022), November 03–05, 2022, Beijing, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3571662.3571666

1 INTRODUCTION

An examination is an indispensable part of the learning process to measure the ability of the students. With many schools and universities have been using an online platform for class and study purposes, the test is also conducted online. While the answers for multiple-choice questions have been graded automatically, the same cannot be said about the answers that need to be written in free text like short answers. Because of that, processing short answer is more complicated than multiple-choice. Thus, this problem falls into the problem of Natural Language Processing (NLP) [1].

The use of Natural Language Processing (NLP) in the field of education has been researched by many in past years. One of them is Automated Short Answer Grading (ASAG). The study of ASAG focused on automating the grading process for the short answer type of question. While the answers for multiple-choice questions have been graded automatically, the same cannot be said about the short answer questions. The short answer is more complicated than the multiple-choice one. Since the input is in the form of text, this problem falls into the problem for Natural Language Processing.

Most of them are still graded manually by the teacher or lecturer which is time-consuming [2, 3]. Besides being more time-efficient, the use of technology to automate the grading process also minimizes the risk of bias or subjectivity [3] as well as inconsistency in grading. Research also shows that the automated grading system can reduce the workload of the teacher [4]. These advantages work for both teacher or lecturer and the student. With faster and more consistent grading on the grader's side, the student can look at their results immediately and the score given is fairer.

The development of ASAG starts from the development of the Automated Essay Scoring (AES) system by Page [5]. Since then, many methods have been proposed to grade essays in various languages. Ishioka and Kameda [6], in their research, proposed an automated grading system for the Japanese language using Latent Semantic Analysis (LSA). LSA is a popular method both for AES and ASAG as it is simple yet has good performance.

More recent research shows that deep learning, such as Long Short-Term Memory (LSTM) is a potential method for grading, as researched by Tulu et.al [7] for the English language, Akmal et.al for the Indonesian language [8] and Oktaviani et.al [9] for the Japanese language. This paper will propose the research using hybrid deep learning CNN-Siamese bidirectional-LSTM based regression method to grade Japanese language examination. The results of the student's answers and the correct answers after going through the process using deep learning were compared using Manhattan

*Corresponding author

ICCIP 2022, November 03–05, 2022, Beijing, China

ACM ISBN 978-1-4503-9710-0/22/11…$15.00
https://doi.org/10.1145/3571662.3571666

distance and the program was written in the Python programming language.

2 LITERATURE REVIEW

The following sub-sections explain the fundamental theory behind our work.

2.1 Artificial Neural Networks (ANN)

The human nervous cell system became one of the development bases of the Artificial Neural Network or ANN. ANN is one of the smallest parts of machine learning and is the basis or core of deep learning algorithms that mimic how human nerves transmit signals between each other. ANN consists of several node layers and output layers. There are 3 parts of an input layer, one or more hidden layers, and one output layer [10].

Deep learning works based on the human nervous system which is a combination of input data, weights, and biases that work together to accurately recognize, classify, and describe objects in data. Neural networks in deep learning consist of several layers that are connected using nodes, where the network is built based on the previous layer to optimize and perfect predictions or categorizations. If this computation always moves forward, it will be called forward propagation. The input and output layers on deep learning networks can be said to be visible layers. This is because the input layer is where the model of deep learning absorbs data for processing while the output layer is the place where the final prediction or final classification is made. The other process is backpropagation which uses several algorithms such as gradient descent.

2.2 Convolutional Neural Network (CNN)

CNN or Convolutional Neural Network is one of the deep learning algorithms that is often used to process images, sounds, or videos. However, CNN can be used in processing and processing input in the form of text. CNN has a significant difference, namely, in one of its layers which is referred to as the convolutional layer, the layer can not only accept input in the form of two-dimensional data but can also receive data in the form of one-dimensional and three-dimensional. Generally, CNN has three main layers in input processing, namely the convolutional layer, pooling layer, and fully connected layer [11].

The convolutional layer is a layer based on convolution, which is a mathematical combination between two relations to produce a third relationship where convolution produces two sets of information. The result of this process can be said as a feature map, activation map, or convolved feature. The pooling layer or commonly referred to as downsampling is the processing of feature maps by reducing dimensional complexity or reducing the parameters of the input used to facilitate the computing process by reducing connections between layers and operating independently for each feature map. In fully connected, this layer does the task as an input classifier based on the features that have been extracted in the previous layer and the filters that have been used so that it outputs the output as desired by CNN.

2.3 Bidirectional Long Short-Term Memory

LSTM or Long Short-Term Memory is a deep learning algorithm that is also a branch of another deep learning algorithm, namely RNN or Recurrent Neural Network. A significant difference from RNN is that LSTM has better memory capabilities than RNN. LSTM was proposed by Sepp Hochreiter and Jurgen Schimdhuber [12] as the solution to the vanishing gradient and exploding gradient that happens in RNN. The architecture of LSTM consists of cell states, hidden states, and several gates. The cell state is just like the memory for the LSTM. The computation and calculation for LSTM are done on the hidden state. The gates: forget gate, input gate, and output gate will determine what information that should be forgotten and what should be passed.

Bidirectional LSTM or BiLSTM is one form of LSTM with a difference in how the direction of the LSTM works. In ordinary LSTM, the direction of using LSTM is only one direction, namely forward or backward so it can cause problems in terms of classification. Thus, BiLSTM overcomes this problem by having input running in both directions and different recurrent nets but the same output layer. This makes BiLSTM able to maintain information that has previously been processed or will be processed.

In hybrid CNN-BiLSTM, the student answer and the answer key will be compared with the Manhattan distance which can be calculated as the sum of the absolute differences between the two vectors. There are many practices on the hybrid algorithm. One of them is the research by Omar Alharbi [13]. The method starts with preprocessing by using the word embedding matrix which is then continued with a convolutional layer to get a feature map. This layer is also useful for extracting features from input text. The next thing to do is to send to the pooling layer with the max pooling feature to reduce the results obtained and get more optimal input. The output of the pooling layer will be sent to the BiLSTM layer to get more optimal results and perform sentence synthesis. Similar architectures are also used in the research by Yue and Li [14] using CNN-BiLSTM and Tam et.al [15] with BiLSTM. Their results show that hybrid models have good performance.

3 THE AUTOMATED GRADING DESIGN

3.1 System Design

This automated scoring system is designed to grade short answers in the Japanese language. In general, the system will compare the answers from students and the answer keys owned by the lecturer using deep learning. Then, from the two answers, it will produce the final value of the student's answers. The answer used is an answer to an essay question so that the predicted value is based on accuracy in answering. The general system design is shown in Figure 1.

The dataset used in this study is obtained from the mid-semester exam at Japanese Studies, Faculty of Humanities Universitas Indonesia. It consists of 5 questions with 43 students' answers for each question. All of them are in the Japanese language, written in either katakana, hiragana, or kanji. This data is far too small for the training process. Therefore, it needs to be augmented. During data augmentation, random words are added to the sentence, or the existing words are replaced by words with similar meanings.

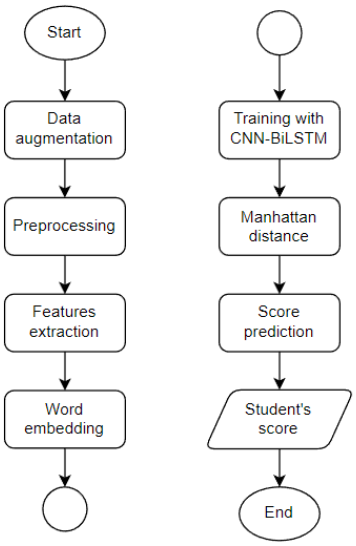

Figure 1: The General Design of The System

3.2 Preprocessing

The augmented data will undergo the next step, preprocessing. Preprocessing is a process where the required data such as student answers and answer keys are processed first before being entered into the system so that the data can be recognized and processed by the machine making it easier for further processing. In this process, the first thing to do is to clean up punctuation marks (such as ,.,?,!,(,)) and whitespace characters (such as " n", "r", "" and others). Then, the sentence will be separated into a word chunks. This process is called tokenization.

3.3 Word Embedding

The next processes are the feature extraction and the word embedding process. During the feature extraction, the results of the data that have been obtained through the preprocessing process will be transformed into vectors that can be measured. Each vector from the text obtained is collected into bi-gram. The number of occurrences of the bi-gram in the student key will be compared to the one in the answer key. Since there are more than one answer keys, this process will select the answer key with the most similarity to the student's answer.

The word embedding process aimed to convert the words into a form that can be understood by the system, which is a vector. The first thing to do is to create a vocabulary that contains unique words in all the answers, which is then followed by the use of padding which is then followed by Word2Vec. In this process, the system will create an embedding array based on the pre-trained word vector from FastText so that if there are words that have similarities with the word vectors, then the matrix index will change according to the available word vectors, and if not found, then the value of the word in the matrix will be 0. The results from the word embedding process will be split into training and testing data. The word vectors representing each number will be processed as input into the CNN-BiLSTM hybrid deep learning model and manhattan distance.

The system will carry out the process in terms of analyzing the level of accuracy between student answers and the lecturer's answer keys, then the system will also create a deep learning instance model for the number of questions and conduct training for each instance. The trained model will be used to process the test data, get a value for each question, and then calculate the final score and error.

3.4 Deep Learning and Manhattan Distance

In short, the use of the Siamese architecture can be used in SIMPLE-O with hybrid CNN-BiLSTM because in Siamese architecture, there are two different inputs so that the two existing inputs can be processed simultaneously. The first input uses student answers while in the second input, the lecturer's answer key becomes the input. The block diagram of CNN-BiLSTM is shown in Figure 2.

The CNN process includes a convolution layer and max pooling layer, both of which are used in a one-dimensional form, which is generally used in a text, which will then go through the BiLSTM process to get more optimal results. The results of the CNN-BiLSTM process will be combined and become input for the Manhattan distance layer which is useful for comparing the distance or difference between the two vectors. The result of the Manhattan distance is a value from 0-1 as the distance between the two inputs. Furthermore, the output will enter the dense layer and produce output in the form of model instances in the form of values that have been classified in the previous layer.

After the model is finished training, the model can be used to predict the value by comparing the test data and the lecturer's answer key. The results from the machine will be compared in the testing process to get predictive results between machine and human assessments. The results of these predictions will provide a value for each question and then will be calculated to get the final score.

3.5 Error Calculation

To know the system's performance, the metric to evaluate the system's results are needed. In this research, the error of the system will be measured by calculating the error rate and Mean Absolute Percentage Error (MAPE). The error of the system, the error rate as shown in (1), and the Mean Absolute Percentage Error (MAPE) as shown in (2) will be calculated using the respective formula.

$$\%\text{EEROR} = \frac{|\ system\ score - human\ rater\ score\ |}{human\ rater\ score} \times 100 \quad (1)$$

$$\%\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \frac{|\ system\ score - human\ rater\ score\ |}{human\ rater\ score} \quad (2)$$

4 RESULTS AND DISCUSSIONS

This section discusses the test results and analysis of the results of the automated essay scoring system. The research was conducted using 7 scenarios to find suitable hyperparameters for the model. For each experiment, 5 instances of deep learning models were generated, and many questions were tested. The results of the engine are compared with the test data and predict the per-question score and the final result is calculated. The test data is 43 data

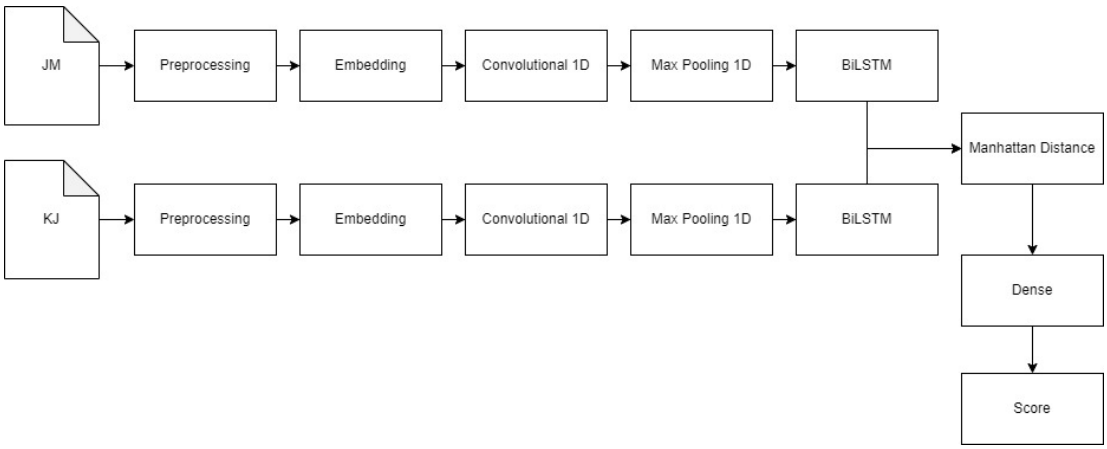

Figure 2: CNN-BiLSTM Process

Table 1: Hyperparameter for Each Tested Scenario

Scenario	Optimizer	Pool Size	BiLSTM Units	Learning Rate	Batch	Epoch
1	Adam	4	25	0.001	64	50
2	RMSprop	4	25	0.001	64	50
3	Adam	4	25	0.001	32	25
4	SGD	2	50	0.01	64	25
5	Adam	2	50	0.001	64	25
6	Adam	4	50	0.001	64	25
7	Adam	4	50	0.001	64	25

Table 2: Results and Errors for Tested Each Scenario

Scenario	Average			Standard Deviation		
	Human Rater	System Score	%Error	Human Rater	System Score	%Error
1	74.81	49.07	36.32	20.82	15.17	14.93
2		46.42	39.08		13.9	21.47
3		45.07	38.33		11.55	16.1
4		40.09	45.95		17.49	21.03
5		44.37	40.53		13.44	17.3
6		44.7	39.89		18.43	20.14
7		54.88	29		17.32	19.34

while the testing data uses 4158 data. Hyperparameter used in each scenario can be seen in Table 1. For all scenarios, a CNN filter or output of 64 and kernel size of 5 was used.

In scenario 1 and 2, with the same other hyperparameters, the optimizer Adam RMSprop are compared. Scenario 3 aims to compare the result if the batch size and epoch are changed while still using the Adam optimizer. Scenario 4 will use different optimizers, pool size, BiLSTM units, and learning rate. Scenario 5 differs in optimizer and learning rate compared to the previous scenario. For scenario 1 until 5, no dropout layer or regularization was used in the experiment, but in scenarios 6 to 7, a dropout layer or regularization are used. Use a regularization or dropout layer to reduce overfitting. The dropout layer is used in scenario 6 and regularizer L1 in scenario 7. Results and error graphs from each scenario are shown in Table 2.

The results of scenario number 1 and 2 show that, in this case, Adam optimizer performs better than RMSprop. This can be seen from the lower average error percentage and the vastly lower error percentage in standard deviation. This finding is similar to [8] where Adam optimizer performs generally better than RMSprop. However, the model in scenario 1 shows the risk of overfitting. This might happen if the epoch is too big. The experiment with scenario 3 is conducted by reducing the epoch. The change of epoch from 50 to 25 managed to reduce the model loss as shown in Figure 3. However, the validation loss is still high.

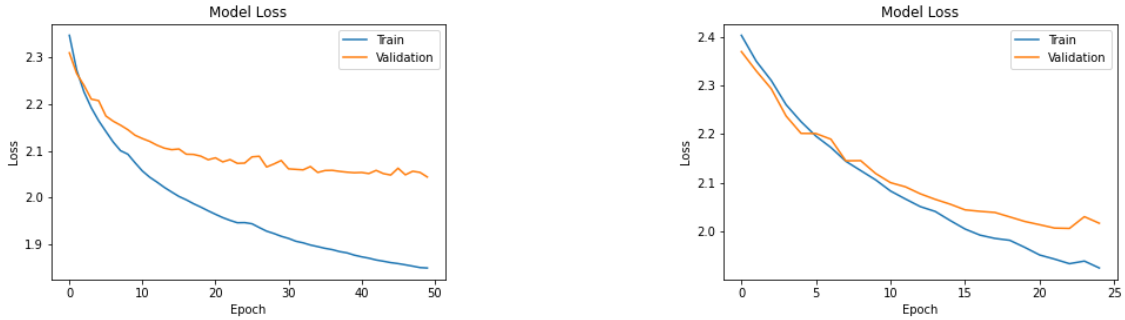

Figure 3: Model Loss Scenario 1 (left) and Scenario 3 (right)

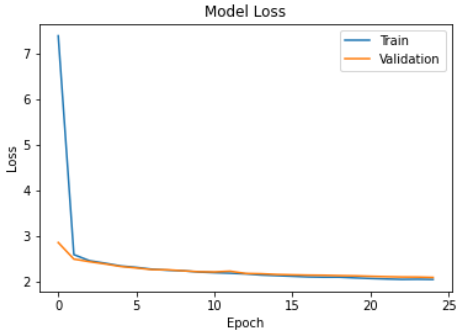

Figure 4: Model Loss Scenario 7

The result for scenarios 4 and 5 have high error percentages and standard deviations. These show that reducing the learning rate and pool size affects the accuracy of the system. A high learning rate might cause the training process to become too fast and not optimal. This high error might also be caused by the higher unit number. When there are too many units, the model can overfit and if there are too few, the model might underfit.

Scenario 6 uses the dropout layer with the value of 0.3 during the training process to reduce overfitting. This layer is added after the pooling layer and before the BiLSTM layer. However, the result shows that the error percentage is still high. The dropout layer's placement and its value need to be set properly to get a better outcome.

Regularization is also one of the methods to reduce overfitting. Regularizer L1 with the value of 0.01 is implemented in scenario 7. This regularizer has a feature selection that is suitable for large training data. The results for the model loss show a good learning curve model as shown in Figure 4. The use of a regularizer reduced the loss and the error percentage of the model.

From these scenarios, it is found that scenario 1 has a lower average error percentage than the other 4 scenarios with an average error percentage of 36.32%, while scenario 7 has a lower error percentage than scenario 6 with an average percentage error of 29%. That can conclude scenarios 1 and 7 are the best results in each category.

For the comparison, scenario 7 has a lower minimum percentage error with 0% error, which can be concluded, for one student the machine score is similar to the human rater's score. But the maximum error percentage error of scenario 1 is lower than scenario 7 with 67.65%. The average error of scenario 7 is also lower than scenario 1 with an average percentage error of 29%. For the system score standard deviation, scenario 7 has a standard deviation of machine value of 17.32, it can be concluded that the results of scenario 7 are close to the standard deviation of human scoring which has a value of 20.82 compared to scenario 1. In addition, it can be concluded, scenario 7 has more diverse results than scenario 1.

The results obtained by this paper have a lower error rate than previous studies written by A. N. Oktaviani, et al.[9] who use CNN and CNN-LSTM in their paper where in that paper, the use of CNN-LSTM gets an average error percentage of 29.93% which is higher than in this paper which is only 29%.

5 CONCLUSION

The experiments show that the most stable model with Deep Learning CNN-Bidirectional LSTM is the Deep Learning Model with hyperparameter kernel sizes value of 5, the number of filters or CNN outputs of 64, pool size of 4, Bidirectional LSTM units of 50, batch size of 64. While not by much, the use of CNN-Bidirectional LSTM can slightly improve the result from the previous study which has an average error percentage of 29.93% to 29%. There is still room for improvement to the model. In the future, the current method will be analyzed and optimized to get more precise predictions.

REFERENCES

[1] Michael Mohler and Rada Mihalcea. 2009. Text-to-text semantic similarity for automatic short answer grading. Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics on - EACL '09 (2009). DOI:http://dx.doi.org/10.3115/1609067.1609130

[2] Pedro Henrique Calais Guerra, Adriano Veloso, Wagner Meira, and Virgílio Almeida. 2011. From bias to opinion. Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11 (2011). DOI:http://dx.doi.org/10.1145/2020408.2020438

[3] Kristen DiCerbo. 2020. Assessment for learning with diverse learners in a Digital World. Educational Measurement: Issues and Practice 39, 3 (2020), 90–93. DOI:http://dx.doi.org/10.1111/emip.12374

[4] Wim Westera, Mihai Dascalu, Hub Kurvers, Stefan Ruseti, and Stefan Trausan-Matu. 2018. Automated essay scoring in Applied Games: Reducing the teacher bandwidth problem in online training. Computers & Education 123 (2018), 212–224. DOI:http://dx.doi.org/10.1016/j.compedu.2018.05.010

[5] Ellis B. Page. 1967. Statistical and linguistic strategies in the computer grading of essays. Proceedings of the 1967 conference on Computational linguistics - (1967). DOI:http://dx.doi.org/10.3115/991566.991598

[6] T. Ishioka and M. Kameda. 2004. Automated japanese essay scoring system:Jess. Proceedings. 15th International Workshop on Database and Expert Systems Applications, 2004. (2004). DOI:http://dx.doi.org/10.1109/dexa.2004.1333440

[7] Cagatay Neftali Tulu, Ozge Ozkaya, and Umut Orhan. 2021. Automatic short answer grading with SEMSPACE sense vectors and malstm. IEEE Access 9 (2021), 19270–19280. DOI:http://dx.doi.org/10.1109/access.2021.3054346

[8] Akmal Ramadhan Arifin, Prima Dewi Purnamasari, and Anak Agung Putri Ratna. 2021. Automatic essay scoring for Indonesian short answers using siamese Manhattan long short-term memory. *2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)* (2021). DOI:http://dx.doi.org/10.1109/icecce52056.2021.9514223

[9] Amanda Nur Oktaviani, Marwah Zulfanny Alief, Lea Santiar, Prima Dewi Purnamasari, and Anak Agung Ratna. 2021. Automatic Essay Grading System for japanese language exam using CNN-LSTM. 2021 17th International Conference on Quality in Research (QIR): International Symposium on Electrical and Computer Engineering (2021). DOI:http://dx.doi.org/10.1109/qir54354.2021.9716165

[10] IBM Cloud Education. What are neural networks? Retrieved December 14th, 2021 from https://www.ibm.com/cloud/learn/neural-networks

[11] IBM Cloud Education. What are convolutional neural networks? Retrieved December 14th, 2021 from https://www.ibm.com/cloud/learn/convolutional-neural-networks

[12] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780. DOI:http://dx.doi.org/10.1162/neco.1997.9.8.1735

[13] Omar Alharbi. 2021. A deep learning approach combining CNN and BiLSTM with SVM classifier for Arabic sentiment analysis. International Journal of Advanced Computer Science and Applications 12, 6 (2021). DOI:http://dx.doi.org/10.14569/ijacsa.2021.0120618

[14] Wang Yue and Lei Li. 2020. Sentiment analysis using word2vec-CNN-BILSTM classification. *2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS)* (2020). DOI:http://dx.doi.org/10.1109/snams52053.2020.9336549

[15] Sakirin Tam, Rachid Ben Said, and O.Ozgur Tanriover. 2021. A convbilstm deep learning model-based approach for Twitter sentiment classification. *IEEE Access* 9 (2021), 41283–41293. DOI:http://dx.doi.org/10.1109/access.2021.3064830