TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 19, No. 3, June 2021, pp. 754~761
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v19i3.18356

Summarization of COVID-19 news documents deep learning-based using transformer architecture

Nur Hayatin, Kharisma Muzaki Ghufron, Galih Wasis Wicaksono
Department of Informatics, Faculty of Engineering, University of Muhammadiyah Malang, Indonesia

Article Info

Article history:

Received Jul 25, 2020
Revised Oct 12, 2020
Accepted Oct 23, 2020

Keywords:

COVID-19
Deep learning
News summarization
Transformer architecture

ABSTRACT

Facing the news on the internet about the spreading of Corona virus disease 2019 (COVID-19) is challenging because it is required a long time to get valuable information from the news. Deep learning has a significant impact on NLP research. However, the deep learning models used in several studies, especially in document summary, still have a deficiency. For example, the maximum output of long text provides incorrectly. The other results are redundant, or the characters repeatedly appeared so that the resulting sentences were less organized, and the recall value obtained was low. This study aims to summarize using a deep learning model implemented to COVID-19 news documents. We proposed transformer as base language models with architectural modification as the basis for designing the model to improve results significantly in document summarization. We make a transformer-based architecture model with encoder and decoder that can be done several times repeatedly and make a comparison of layer modifications based on scoring. From the resulting experiment used, ROUGE-1 and ROUGE-2 show the good performance for the proposed model with scores 0.58 and 0.42, respectively, with a training time of 11438 seconds. The model proposed was evidently effective in improving result performance in abstractive document summarization.

Corresponding Author:

Kharisma Muzaki Ghufron
Department of Informatics
University of Muhammadiyah Malang
Campus III UMM, 246 Raya Tlogomas St., Malang, Jawa Timur 65144, Indonesia
Email: kharisma.muzaki@webmail.umm.ac.id

1. INTRODUCTION

In early 2020 the world was hit by Corona virus disease 2019 (COVID-19) pandemic, which affected global life. All people with various backgrounds and fields of science discuss COVID-19 pandemic, both through social media and web news. Likewise, with data scientists, several studies primarily related to natural language processing (NLP) on COVID-19 also began to be carried out. Such as predictive models that can estimate returns on stocks from countries most affected by the COVID-19 pandemic [1], also modeled causality using neural networks to explore misinformation on social media during the COVID-19 pandemic [2]. As far as our observation, none of these studies related to the COVID-19 news document has been done previously. Meanwhile, one of the news media trends recently is the publication of news documents about COVID-19, which was released quickly and updated every day, so that the release resulted in a lot of data and news about the COVID-19. This was proofed by the search results from the Google search engine that gave a lot of output above 6 billion documents for COVID-19 keywords that are accessed in July 2020. This is a challenge in how

to be able to find the relevant information from the document collection, especially from the news documents. News documents are one of the many unstructured data that are found and easily accessed on the internet. Multiple types of data that are scattered on the internet, including news documents, increase rapid growth [3, 4] and increase exponentially [4]. With the rapid growth of data, the summarization of information becomes important to meet the needs of internet users. In summarizing documents, two types can be used, namely abstractive and extractive. An extractive summary is a concise text method which consists of three stages: text representation, sentence evaluation, and sentence selection using a statistical model [5]. Abstractive summarization works by producing new sentences in summary based on an existing text by repeating new words as an extraction act [6].

Several studies on abstractive news summaries using deep learning have been done previously. Abstractive summarizations were conducted on the Chinese news dataset using public opinion [7]. Meanwhile, other research focused on the keywords that exist in the text sentences that can work effectively to produce interpreted texts [8]. The other model that was used for abstractive news summary is sequence modeling, such as long short-term memory (LSTM) and recurrent neural network (RNN). The sequence to sequence the RNNs model has successfully reduced the training loss for abstractive summary used amazon fine food reviews dataset [9]. Unfortunately, the maximum output of long text provides incorrectly. The research provides a correct summary only for short text. Another study also conducted experiments by doing a combination of local attention and LSTM in which the results of the summarization of the characters repeatedly appeared so that the resulting sentences were less organized and the recall value obtained was low [10]. However, the repetitive workings found in the recurrent model like RNN and LSTM, prevent the model from conducting parallel training and limit the ability to know context with longer input sequences [11].

Transformer, as base language models, has significantly impacted the NLP research field to replace the deficiency of both LSTM, CNN and RNN based as a deep learning architecture [12, 13], so that many reasons why the transformer was chosen as base model architecture. Various studies applied to transformer architecture have been carried out and have improved results significantly in document summarization [14]. In previous studies, transformers was used as a detection irony grouping in Spanish using pre-training Twitter word research results compared to LSTM attention, and the deep averaging network showed an increase significantly on performances [15]. The transformer also succeeded in making Chinese story-generation by creating two layers of self-attention and reducing the number of encoder and decoder layers to identic one. The results showed a low loss and an increase significantly from the base layer of the transformer model [16]. Meanwhile, the use of the transformer was carried out successfully using a combined modification of the bidirectional encoder representations from transformers (BERT) as a transformer-based encoder and decoder in Japanese abstractive summarization, which has resulted in good average accuracy and the lowest loss value [17]. In this study, we propose a transformer-based model to summarize COVID-19 news with several methods and stages. Another discussion from the result is present by make sublayer modifications to determine the effect of parameters on the encoder and decoder layers.

2. DATASET

The dataset we used came from news documents about COVID-19 that was published on the Kaggle platform [18] from the Canadian broadcasting corporation (CBC) news site, with a total number of documents that were used to build the model is 2755 documents. The relevance of the news in the dataset containing variations combined topics related to COVID-19 are processed using the crawler with the keyword COVID-19 published from January 08, 2020, until March 03, 2020. In the dataset, there are text description contains the news content, and the description feature is a summary of the news content. Specific keywords in news content are listed in the word coronavirus. There is a variety of mixed news content, but the news content is more important in the amount of COVID-19 growth in each region.

3. PREPROCESSING

Several previous studies have shown the results of their research by doing preprocessing can increase accuracy results by a percentage of 2% [19], preprocessing is also used in some words that have the form of misspellings [20]. At the preprocessing stage, several processes occur. i.e., contractions, lowercasing & printable checks, splitting data, tokenization, and word embedding. We divide the dataset into three forms, i.e., training, validation, and testing, with percentages of 70%, 10%, and 20%, respectively. Contractions and printable checks, mapped out contraction words from the list of word contractions. These words were defined by ourselves to get the original form terms such as "don't" become "do not," then printable check used to delete characters other than punctuation marks, and ASCII letters. After that, due to memory limitations we did distribute control to limit text which was needed as a training model. Furthermore, tokenization is the process

of breaking text into separate words and adding unique tokens. In the modification of the model, we use a pre-trained word embedding global vector (GloVe) with a vocabulary of 2.2 M to present each word in a 300-dimensional vector size [21]. Previous studies have shown that unsupervised comparison results based on text summarization using word embedding are more effective than using a bag of words [22].

4. TRANSFORMER MODEL

We analyzed in Figure 1, and there are encoder and decoder layers that have the dropout and Normalization in each sublayer. We use of gaussian error linear unit (GELU) in the feed-forward network is used only once on each encoder or decoder layer, due to GELU has high complexity in the NLP field but the performance produced is superior compared to other activation functions such as ELU and ReLU [23]. The multi-head attention formulation can be seen following in (1) [14]. h is the total attention carried out in parallel so that every $head_i$ is carried out the attention function contained in (2) [14].

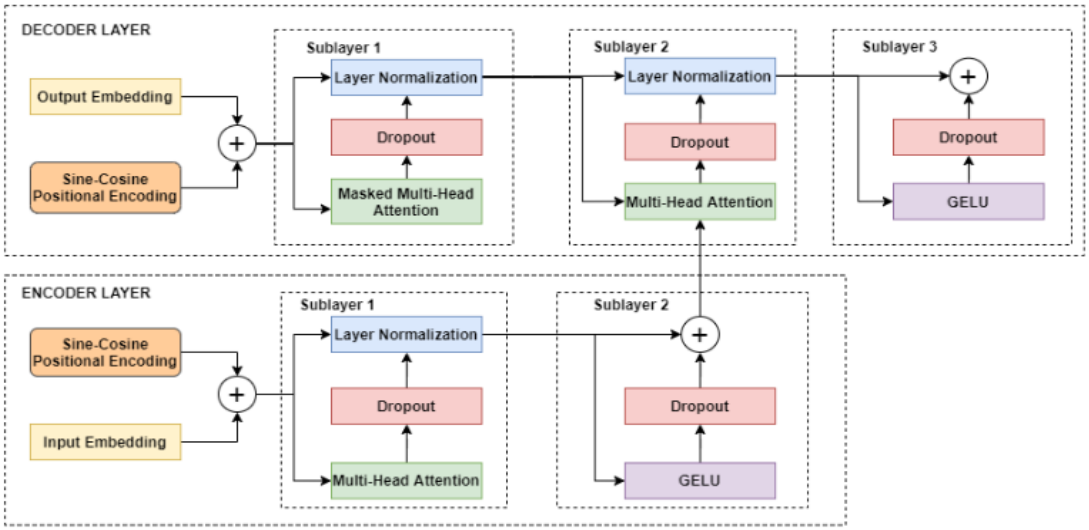


Figure 1. Transformer model proposed

$$MultiHead(Q,K,V) = Concat(head_1,\dots,head_h)W^O \quad (1)$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (2)$$

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}}) \quad (3)$$

The attention function can be defined as a function that performs the mapping of the query Q is the target sequence; the key pair K and the value V are derived from the sequence. Each Q, K, V, and output mapping are defined in vector form. The weight of each calculated value is a representation of adjusting the query to the key. The query and key dimensions are defined as d_k, and the values dimension d_v is used as the Attention parameter found in (3) [14]. Multi-head attention combines several attention models to each of the Q,K,V models. The weighting dimension of a sequence is defined as d_{model} so that its representation is in multi-head $W^O \in \mathbb{R}^{d_{model} \times d_k}$. The primary difference between a masked multi-head attention and a multi-head attention is that some tokens contained in a sequence are randomly removed to train the model to understand the context contained in the sequence. Transformer also performs positional encoding (PE), which is the injection of some information on each word position contained in a sequence. PE has the same dimensions as d_{model}. In this paper, we use sine-cosine positional encoding, where the formula equation can be seen in (4) and (5), the *pos* is a position, and *i* is dimension.

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \quad (4)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \quad (5)$$

Dropout reduces the loss value during the training process, also helps prevent overfitting [24]. The normalization layer normalized values come from the hidden layer. Perform on small batch sizes dependent to reduce memory cost, normalization can rely for increase training accuracy [25]. The normalization layer minimize parameter change during propagated through the deep networks [26].

5. SCORING

The summarization result measurements are performed using recall-oriented understanding for gisting evaluation (ROUGE) [27]. We chose the ROUGE-N method, which was represented in (6), in which the calculation is based on n-gram recall [27]. Where *n* is the length of n-gram, *Ref* is a set of reference summaries. $Count_{match}(gram_n)$ is the calculation of the maximum number of n-grams co-occurring on the generated summaries model and the set of reference summaries. $Count(gram_n)$ is the number of n-grams in reference summaries.

$$ROUGE - N = \frac{\sum_{S \in Ref} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in Ref} \sum_{gram_n \in S} Count(gram_n)} \quad (6)$$

6. EXPERIMENT AND DISCUSSION

In accordance with Figure 2, the experiment from all transformer summarization models that we build first performed preprocessing on the text. The use of preprocessing is to reduce less relevant features, and the amount of memory needed to carry out the training process [28]. Then after preprocessing, we perform all models transformer-based architecture with encoder and decoder that can be done several times repeatedly and make a comparison of layer modifications based on scoring.

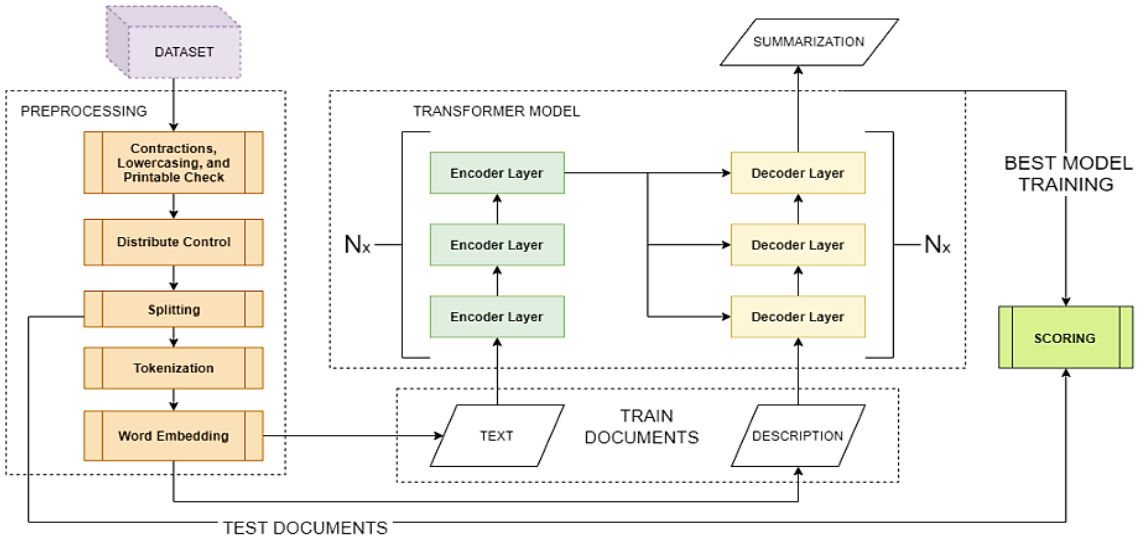


Figure 2. The news summarization experiment of COVID-19

6.1. Software and splitting

The specifications in the experiment used Google colab cloud computing with data as follows: Intel Xeon CPU 2.20GHz, 14GB RAM, Tesla P100 16GB GPU. The Python TensorFlow library is used as a deep backend learning where calculations are performed on the GPU. The results of splitting on the COVID-19 news document dataset were 1928 documents to conduct training, 275 documents as validation during the training process, and 552 documents to test the results of the training model, for each training model. Validating perform under a batch of 32 documents to calculate the average ROUGE-1 score and loss value. The maximum length of the news text content of the entire training document was equal to 600 and 25 in the text description.

6.2. Experiment scenario

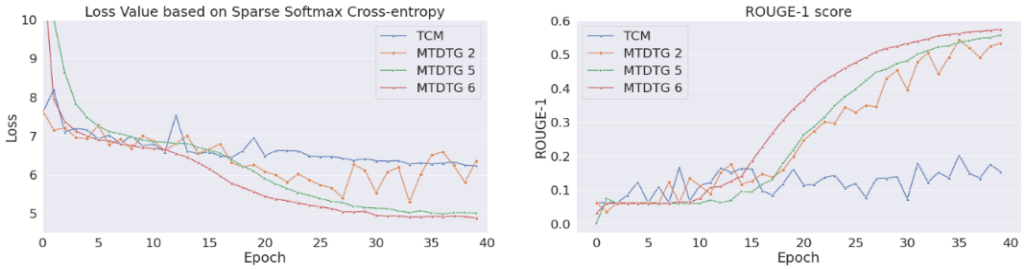
In this research experiment, we use Adam which is the stochastic based optimization method to update the weight value of the loss value measurement results [29], where the calculation of the loss value of the weight value used sparse softmax cross-entropy. Table 1 shown the experiment scenario of several different parameters used to build the transformer deep learning model. Adam optimization parameters used are beta1=0.9, beta2=0.98, epsilon=1e-9. All models had carried out fairly iterations of 40 epochs on training experiments. From the existing transformer design model in previous studies, we chose the transformer C model (TCM) [14] as a comparison test with some of the models that we proposed. The selection is because TCM has the most straightforward design and the results of trials with other architectures that are more complex by 1%. The model that we proposed includes tokenization and word embedding, and in the form of parameter changes modification of the encoder-decoder layer, or can be called a modified base model transformer with distribute control tokenization and GloVe word embedding. The postfix number is a representation of the number of identical layer encoder decoders (MTDTG Nx). The activation function in the MTDTG model which was used in this research is the GELU function to calculate the weight of the sequence in the feed-forward layer, whereas in previous studies using ReLU as an activation function.

Table 1. Scenario experiment

Parameter	TCM [14]	Our proposed models		
		MTDTG 2	MTDTG 5	MTDTG 6
heads	8	10	8	10
learning rate	1e-3	1e-3	1e-5	1e-2
node feed-forward	256	512	256	512
dropout rate	0.1	0.2	0.1	0.2
attention dropout rate	0.1	0.2	0.1	0.2
encoder layer	2	2	5	6
decoder layer	2	2	5	6
activation function	ReLU	GELU	GELU	GELU

6.3. Experiment result

During the training process, a loss value and ROUGE-1 is obtained, as shown in Figure 3. TCM has decreased loss in epoch 40, so that a loss value of 6.3 is obtained. In other models ranging from epoch 18-27 loss in MTDTG 2 gradually decreased at 5.5, then climbed back up because of the variant batch in the document vary, so that the model must recognize new data again. Other models as shown in Figure 3 (a), the graph depicts that the MTDTG 5 decreased gradually in loss at epoch 10 and 25 with obtained loss value of 4.8. The latest results of our experiment on MTDTG 6 have decreased a loss value to 4.7 which is not much different compared to MTDTG 5 with a difference of 0.1%. During the training phase, each epoch validation based on maximum ROUGE-1 score is considered to save the weight model. This experiment used ROUGE-1 to summarization result measure, the result of ROUGE-1 was shown by Figure 3 (b). The graph explained the validation process where each epoch TCM has a maximum score 0.20. Our proposed model shown by MTDTG 2, MDTG 5, and MTDTG 6, those models have an outperformed result measured by ROUGE-1 with score 0.54, 0.59, and 0.60 respectively.



(a) (b)

Figure 3. Comparison; (a) model training loss and (b) ROUGE-1 score validation

6.4. Summarization model

We try to explore the results of MTDTG 6, on the word cloud, as seen in Figure 4. Word cloud describes the word frequency representation of the whole document from the generated summarization conducted by MTDTG 6. Total all unique words while performing summarization appeared in Figure 4 is 200

words that used the MTDTG 6 model. Most word generated on summarization is 'say', this result is caused by the dataset used related to news content to get information from an interview on experts or government officials. The word "Canadian" also appeared, because this word related to the news that has been reported mostly on Canadian during the coronavirus situation.

We classified keyword based on frequent of appearance. On Table 2, there are two types of keyword: "most" (keywords have a high frequent), and "least" (keywords appeared rarely). The percentage of overall word cloud summarization results from Figure 4 is seen the table. The most 5 words that have high an appear frequency are "say, will, coronavirus, people, pandemic". These keywords prove that the main topic of news reported widely during this period is about the corona virus pandemic. It caused the government of Canada to take policies focused on public health which concerned with the safety of citizens. The other hand, the fewest 5 words found: "families, disease, grocery store, first nation, stock market". The main thing that had been less concern from report news is that the coronavirus pandemic affected the weak of many sectors, including trading and economy, and also less reported about the policy issue to close public places i.e. park and store.

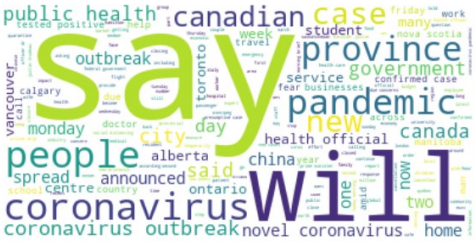


Figure 4. Word cloud summarization MTDTG 6

Table 2. Most and least word representation

Type	Rank	Keywords	Percentage
most	1-5	say, will, coronavirus, people, pandemic	0.83
	6-10	province, new, case, canadian, coronavirus outbreak	0.56
	11-15	public health, government, city, canada, said	0.45
	16-20	day, monday, spread, novel coronavirus, two	0.38
	21-25	outbreak, china, announced, home, now	0.34
least	196-200	disease, families, masks, major, caused	0.08
	191-195	death, threat, operations, warning, place	0.08
	186-190	price, potential, store, nurse, park	0.08
	181-185	provincial health, care workers, self isolation, slow spread, stock market	0.08
	176-180	first nation, grocery store, expert, public, including	0.09

6.5. Testing model

The distribution of ROUGE-1 and ROUGE-2 scores on MTDTG 6 is shown in Figure 5, the number of summarization results is at most 80-100% with a total of 209 documents, while at least 10% is scored with a total of 33 documents. Inversely proportional to ROUGE-2, where the most distribution in the result of the score is in the 0-20% because ROUGE-2 uses the bigram mechanism. So, there are several overlapping words between summarization and references. The comparison results test also shown in Table 3. The results of MTDTG 5 and MTDTG 6 on the ROUGE-1 score have a difference of 1% when used on the ROUGE-2 score. There is quite a difference of 3%. The evaluation results show that MTDTG 6 with a ROUGE-1 score of 0.58 and a ROUGE-2 score of 0.42 is the best model that can be used in the test data.

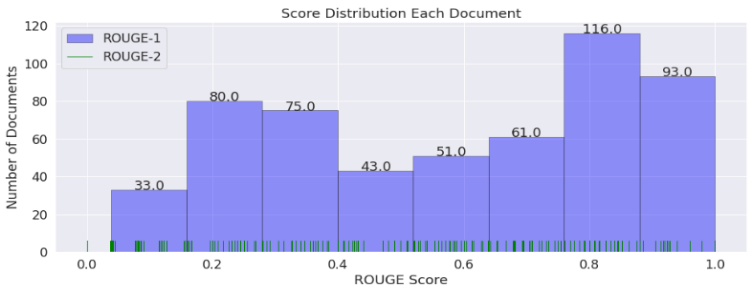


Figure 5. Distribution ROUGE score on test documents

From the results of the entire trial of the test document, there is only MTDTG 6 architecture that can be supplied with the maximum due to the memory limitations handled by the GPU. In the experiment, the researcher often got constraints on out of memory (OOM) resources. This obstacle can be overcome by reducing the architectural design model, especially the most important things, i.e., the number of feed-forward networks, batch size, and the number of encoder-decoders. The disadvantage of MTDTG 6 is that the memory needed to conduct training is more significant because we use 300-dimensional GloVe as word embedding. However, the results of the test model can increase by a percentage of 13% in ROUGE-1 and 16% ROUGE-2 compared to TCM.

Table 3. Overall comparison score model

Model	Validation	Test		Training time (second)
	Maximum ROUGE-1	ROUGE-1	ROUGE-2	
TCM [14]	0.20	0.45	0.26	**1723**
MTDTG 2	0.54	0.51	0.34	5046
MTDTG 5	0.59	0.56	0.38	7606
MTDTG 6	**0.60**	**0.58**	**0.42**	11438

7. CONCLUSION

Summarization of news documents COVID-19 based on deep learning using transformer architecture can be done by compiling various models and methods of activation functions. We proposed the transformer with architectural modification as the basis for designing the model in abstractive document summarization, which was evidently effective in improving result performance. The best model MTDTG 6 performs that was measured using the ROUGE-1, and ROUGE-2 has obtained a good score of 0.58 and 0.42, respectively, with a training time of 11438 seconds. Based on word clouds from all documents with the most discussion, we found that the most reported in news related to the policy and regulation from the government on public health services prioritization during COVID-19 pandemic. Since the research of COVID-19 news document abstractive summaries is minimal, many research opportunities can be done further by modifying the encoder and decoder layer to get better model quality results, and they also work with faster training time. The integration of several transformer architecture models can also be done, such as the use of the T5 or BART models to summarize until the quality of existing research can be evaluated or compared.

REFERENCES

[1] A. A. Salisu and X. V. Vo, "Predicting stock returns in the presence of COVID-19 pandemic: The role of health news," *Int. Rev. Financ. Anal.*, vol. 71, 2020.
[2] A. K. M. N. Islam, S. Laato, S. Talukder, and E. Sutinen, "Misinformation sharing and social media fatigue during COVID-19: An affordance and cognitive load perspective," *Technol. Forecast. Soc. Change*, vol. 159, 2020.
[3] A. Khan and N. Salim, "A review on abstractive summarization methods," *J. Theor. Appl. Inf. Technol.*, vol. 59, no. 1, pp. 64-72, 2014.
[4] M. Marjani, *et al.*, "Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges," *IEEE Access*, vol. 5, pp. 5247-5261, 2017.
[5] T. Uçkan and A. Karcı, "Extractive multi-document text summarization based on graph independent sets," *Egypt. Informatics J.*, vol. 21, no. 3, 2020.
[6] S. Gupta, *et al.*, "Abstractive summarization: An overview of the state of the art," *Expert Syst. Appl.*, vol. 121, pp. 49-65, 2019.
[7] Y. Huang, Z. Yu, J. Guo, Z. Yu, and Y. Xian, "Legal public opinion news abstractive summarization by incorporating topic information," *Int. J. Mach. Learn. Cybern.*, vol. 11, pp. 2039-2050, 2020.
[8] C. Yuan, Z. Bao, M. Sanderson, and Y. Tang, "Incorporating word attention with convolutional neural networks for abstractive summarization," *World Wide Web*, vol. 23, no. 1, pp. 267-287, 2020.
[9] A. K. Mohammad Masum, *et al.*, "Abstractive method of text summarization with sequence to sequence RNNs," *2019 10th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2019*, 2019, pp. 1-5.
[10] P. M. Hanunggul and S. Suyanto, "The Impact of Local Attention in LSTM for Abstractive Text Summarization," *2019 2nd Int. Semin. Res. Inf. Technol. Intell. Syst. ISRITI 2019*, pp. 54-57, 2019
[11] B. Myagmar, J. Li, and S. Kimura, "Cross-Domain Sentiment Classification with Bidirectional Contextualized Transformer Language Models," *IEEE Access*, vol. 7, pp. 163219-163230, 2019.
[12] Y. Chen and H. Li, "DAM: Transformer-based relation detection for Question Answering over Knowledge Base," *Knowledge-Based Syst.*, vol. 201-202, 2020.
[13] T. A. Fuad, M. T. Nayeem, A. Mahmud, and Y. Chali, "Neural sentence fusion for diversity driven abstractive multi-document summarization," *Comput. Speech Lang.*, vol. 58, pp. 216-230, 2019.
[14] A. Vaswani, *et al.*, "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 2017, pp. 5999-6009, 2017.
[15] J. Á. González, L. F. Hurtado, and F. Pla, "Transformer based contextualization of pre-trained word embeddings for irony detection in Twitter," *Inf. Process. Manag.*, vol. 57, no. 4, 2020.
[16] J. W. Lin, Y. C. Gao, and R. G. Chang, "Chinese Story Generation with FastText Transformer Network," *1st Int. Conf. Artif. Intell. Inf. Commun. ICAIIC 2019*, pp. 395-398, 2019.

[17] Y. Iwasaki, A. Yamashita, Y. Konno, and K. Matsubayashi, "Japanese abstractive text summarization using BERT," *Proc. - 2019 Int. Conf. Technol. Appl. Artif. Intell. TAAI 2019*, 2019.
[18] Ryan Han, "COVID-19 News Articles Open Research Dataset | Kaggle," 2020. [Online]. Available: https://www.kaggle.com/ryanxjhan/cbc-news-coronavirus-articles-march-26 (accessed Jul. 22, 2020).
[19] L. Ruhwinaningsih and T. Djatna, "A Sentiment Knowledge Discovery Model in Twitter's TV Content Using Stochastic Gradient Descent Algorithm," *TELKOMNIKA Telecommunication Computing Electrical Electronics and Control*, vol. 14, no. 3, pp. 1067-1076, 2016.
[20] M. A. Fauzi, R. F. N. Firmansyah, and T. Afirianto, "Improving sentiment analysis of short informal Indonesian product reviews using synonym based feature expansion," *TELKOMNIKA Telecommunication Computing Electrical Electronics and Control*, vol. 16, no. 3, pp. 1345-1350, 2018.
[21] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," 2014. [Online]. Available: https://nlp.stanford.edu/projects/glove/.
[22] N. Alami, M. Meknassi, and N. En-nahnahi, "Enhancing unsupervised neural networks based text summarization with word embedding and ensemble learning," *Expert Syst. Appl.*, vol. 123, pp. 195-211, 2019.
[23] D. Hendrycks and K. Gimpel, "Gaussian Error Linear Units (GELUs)," *Cornell University,* pp. 1-9, 2016. [Online]. Available: http://arxiv.org/abs/1606.08415.
[24] B. D. Satoto, I. Utoyo, R. Rulaningtyas, and E. B. Khoendori, "An improvement of Gram-negative bacteria identification using convolutional neural network with fine tuning," *TELKOMNIKA Telecommunication Computing Electrical Electronics and Control*, vol. 18, no. 3, pp. 1397-1405, 2020.
[25] E. Gibson *et al.*, "NiftyNet: a deep-learning platform for medical imaging," *Comput. Methods Programs Biomed.*, vol. 158, pp. 113-122, 2018.
[26] I. Nurhaida, V. Ayumi, D. Fitrianah, R. A. M. Zen, H. Noprisson, and H. Wei, "Implementation of deep neural networks (DNN) with batch normalization for batik pattern recognition," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 2, pp. 2045-2053, 2020, doi: 10.11591/ijece.v10i2.pp2045-2053.
[27] C. Y. Lin, "Rouge: A package for automatic evaluation of summaries," *Proc. Work. text Summ. branches out (WAS 2004)*, no. 1, 2004, pp. 25-26.
[28] S. A. Alsaidi, A. T. Sadiq, and H. S. Abdullah, "English poems categorization using text mining and rough set theory," *Bull. Electr. Eng. Informatics*, vol. 9, no. 4, pp. 1701-1710, 2020.
[29] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, 2015, pp. 1-15.

BIOGRAPHIES OF AUTHORS

Nur Hayatin is a lecturer at the University of Muhammadiyah Malang. She received her Master in Informatics Engineering from the Institute of Technology Sepuluh Nopember Surabaya, Indonesia, with an area of interest in data science, where she teaches courses related to Natural Language Processing. Her main research interest is text analytics, social media analytics, data mining, and information retrieval. Email: noorhayatin@umm.ac.id

Kharisma Muzaki Ghufron is currently completing a Bachelor's degree from the Informatics Department, Faculty of Engineering, at the University of Muhammadiyah Malang, Indonesia. His interests include natural language processing and deep learning architecture. Email: kharisma.muzaki@webmail.umm.ac.id

Galih Wasis Wicaksono is a lecturer in the Informatics department at the University of Muhammadiyah Malang, Indonesia. He teaches logic & computing and computer reasoning courses with an area of interest in data science. The focus of his research is case-based reasoning and online learning. Email: galih.w.w@umm.ac.id