# Single Document Summarization Using BertSum and Pointer Generator Network

Rini Wijayanti, Masayu L. Khodra, Dwi H. Widyantoro

School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
rini.wijayanti@students.itb.ac.id, masayu@informatika.org, dwi@stei.itb.ac.id

*Abstract:* The rapid development of textual data requires an automated text summarization system to obtain shortened versions of documents quickly and accurately. This paper investigates the performances of BertSum and Pointer Generator Network (PGN) on the IndoSum corpus containing Indonesian news articles. We compare these methods to NeuralSum, which is claimed to outperform other methods when working with the IndoSum dataset. In our experiment, BertSum with Indonesian's pre-trained model outperformed NeuralSum in extractive summarization. NeuralSum, on the other hand, tends to select the leading sentences as a summary and occasionally produces a blank summary. Meanwhile, PGN effectively prevents word repetition by using a coverage mechanism, although the summary results are sometimes out of context.

*Keywords*: text summarization, deep learning, Bahasa Indonesia, transfer learning, IndoSum

## 1. Introduction

The development of the digital era has produced large amounts of textual data that makes us more difficult to get relevant information due to information overload. Some people also find it challenging to understand the information, so they need more time to extract it manually. Therefore, how to quickly refine the core information has become an urgent problem to solve. Today, various tasks also certainly require a text summarization system, such as summarizing news, e-mails, scientific publications, discussion forums, and generating headlines, snippets, book synopses, biographies, etc.

The automatic text summarization aims to get a shortened version of documents by filtering the most important information from one or several text sources [1]. There are two techniques in summarizing text: extractive and abstractive. Extractive summarization produces a summary by combining the salient of the text unit without changing the original words and sentences. Meanwhile, abstractive summarization creates a summary by rewriting or paraphrasing the text. In this study, we evaluate both techniques applied to the Indonesian news article.

Research in this field began by Luhn in 1958, which creates an abstract from technical documents [2]. Since then, many researchers have made various attempts to improve the quality of summarization results. This topic has gained much attention in the NLP community and is considered a challenging task since computers lack human-like knowledge and language processing ability. In 2014, Kageback showed that continuous vector representation based on neural networks provided better summarization results than conventional approaches [3]. This research becomes the starting point for the development of neural network-based text summarization researches. Instead of doing handcrafted feature engineering, it only requires low-level features as input and thus allows research in low-resource language to be explored.

Research on summarizing Indonesian text using a deep neural network model had started by Kurniawan and Louvan in 2018 [4]. They constructed IndoSum, a new benchmark dataset for Indonesian single text summarization. Tthe best of the author's knowledge that IndoSum, which has roughly 19K articles, is the only large Indonesian summarization corpus available to the public. Several methods have evaluated this corpus, and NeuralSum outperforms other

Received: December 6th, 2020. Accepted: December 20th, 2021
DOI: 10.15676/ijeei.2021.13.4.10

methods significantly [4], [8]. However, NeuralSum still has some weaknesses, i.e., it tends to select the leading sentence as the summary and sometimes produces an empty summary. In the process of sentence selection, it first uses a binary decision of whether to include a sentence as a candidate summary or not. Then, the summary is made based on the sentence position in the document. Thus, sentences in the higher position are likely to be included in the summary. NeuralSum also does not have a mechanism to check sentence redundancy, so it never knows whether another instance is already in the summary candidates or not.

In this study, we also use the abstractive method since the IndoSum gold summary is abstractive. As a comparison to NeuralSum, we employed BertSum and the Pointer Generator Network (PGN) . We are interested in BertSum as it introduced aarchitecture for summarizing the text. BertSum is a transfer learning-based text summarization method that utilizes the BERT's pre-trained model to summarize text both extractively and abstractively. this concept address shortage sBertSumlanguage. This method uses Trigram Blocking in sentence selection to avoid sentence redundancy while avoiding blank summaries. We also considered as it has a coverage method to the repeated text that usually appears in natural language generationOne of these methods uses transfer learning to find the possibility of being used in low-resource language, in this case Indonesian.

This paper aims to investigate whether BertSum and PGN perform better than NeuralSum on the IndoSum corpus.

Wddifferently from the original paper [4] by eliminating the folds and setting the new training, evaluation, and testing ratios One of these methods uses transfer learning to find the possibility of being used in low-resource language, in this case Indonesian.We observed how these three architectures performed when given the Indonesian news articles, and we evaluated the generated summary to its abstractive summary.

We organized this paper as follows: the related works are given in Section II after this introduction. Then, it describes summarization methods of Indonesian text and three other methods we investigated in this study. Next, the evaluation is performed in Section III. Finally, the paper is ended with a conclusion and future works in Sec. IV.

## 2. Related Works

Automatic text summarization (ATS) is a challenging task because the computer has to understand human language. Therefore, it needs a lot of training data and linguistic resources to develop a good understanding of the text being read. This task becomes more challenging in the low-resource language since the dataset is limited or unavailable.

### *A. Indonesian Text Summarization*

Research on Indonesian text summarization also suffered from limited data and other linguistic resources. As a result, many researchers used their dataset, making it difficult to compare the results with other studies. However, an attempt to create a public summarization corpus for the Indonesian language has been done [4], [10]. While F.Koto [10] developed a summarization corpus from Whatsapp conversations, Kurniawan and Louvan [4] created the IndoSum dataset containing online news. Both were created the corpus along with its extractive and abstractive summaries. So far, the IndoSum is the largest publicly available corpus of Indonesian single-document summarization. The availability of this large dataset has driven the use of deep neural networks in summarizing Indonesian text.

Before 2019, most methods used in Indonesian text summarization were unsupervised and non-neural networks. They adopted a linear weighting model to weigh each text unit according to certain features. Therefore, researchers tend to focus on feature optimization. In 2017, Budhi et al. [11] made extractive summaries using the Weighted Directed Graph method for a single article summarization task. They used word similarity to article keywords, word frequency, word position, and the relationship between sentences as text features. Then Herdiyeni et al. [12] used Genetic Algorithm (GA) to get the best weight on the eleven text features used in

text summarization. GA is also used by Silvia et al. [13] with LDA to obtain sentence feature weights. Finally, Tardan et al. [14] used WordNet Indonesia to get semantic similarities between sentences. Besides considering the sentence position, they also calculate semantic vectors.

In 2017, Gunawa et al. [15] performed text summarization using the TextTeaser algorithm. This algorithm uses features such as title features, sentence length, sentence position, and keyword frequency. In the same year, Hutama et al. [16] summarized the text differently by applying 5W + 1H to select sentences that are considered essential. These elements were used as labels for each sentence, and each sentence can have more than one label. They classified sentences using the Naïve Bayes method with 12 features, then sentences containing 5W + 1H were selected as summary sentences. Sabuna et al. [17] also used the machine learning method, Decision Tree, to classify the important sentences, while Maryana et al. [18] only used term frequency to decide which sentence to choose as a summary.

Recently, research on Indonesian text summarization has started to apply the deep neural network (DNN). The DNN-based text summarization methods generally use the following pipelines: (i) creating word embedding vectors, (ii) using word embeddings to encode sentences/documents, and (iii) representing sentence/document (sometimes also word embedding) to create the models [19]. Some researchers have used IndoSum to evaluate their proposed method. For example, Chandraseta and Khodra [8] used the Bi-directional Gated Recurrent Unit (Bi-directional Gated Recurrent Unit) architecture to construct extractive summaries. They formulated this summarization task as a sequence labeling problem by using IndoSum to train and test the developed model. This model was used to update paragraphs from a person's biographical articles on Wikipedia. However, the ROUGE value obtained is not better than the NeuralSum model implemented in [4].

Furthermore, the BiGRU architecture and IndoSum dataset are also used by Halim et al. [20]. They developed a model consisting of 2 layers, where the first layer runs at the word level, and the second layer runs at the sentence level. The results also showed lower than NeuralSum. Unlike the two previous studies, Cai et al. [21] used the clustering method to test the IndoSum. Although the results improved compared to other methods, the paper did not clearly show the clustering method.

### *B. Deep Neural Network-Based Text Summarization*

This paper investigated DNN-based text summarization methods, i.e., BertSum [6] and Pointer Generator Network (PGN) [7], when given the IndoSum dataset. We compared these methods with NeuralSum, which performed well on [4]. NeuralSum, BertSum, and PGN were initially used to summarize English texts, but we evaluate them with the Indonesian news article.

In this study, we compare three deep neural networks and attention-based summarization methods, i.e. NeuralSum [11], Pointer Generator Network [12], dan BertSum [8].

**NeuralSum** [5] is an extractive summarization method developed with encoder-decoder architecture. The encoder is regarded as a document reader by using CNN and LSTM. CNN works at the word level with the max-pooling operation and uses multiple kernels with different widths to obtain several sentence vectors. Then all these sentence vectors are summed to get the final sentence representation. Finally, the sentence embedding from CNN is forwarded to the LSTM to get a document representation to capture local and global information. This illustration is depicted in Fig.1. The figure also shows extracting sentences in the decoder with an attention mechanism to select important sentences. The sentence extractor uses LSTM to label the sentence sequentially by considering its relevance and redundancy with other sentences. The labeling process utilizes document representation as well as labeled sentences in the previous position. Given sentence vectors generated from CNN $(s_1, \dots, s_T)$ and encoder's hidden states from LSTM $(h_1, \dots, h_m)$, then decoder's hidden states $(\bar{h}_1, \dots, \bar{h}_m)$ are calculated by the equation:

$$\bar{h}_t = LSTM(p_{t-1}s_{t-1}, \bar{h}_{t-1}) \quad (1)$$

$p_{t-1}$ is the decoder probability of extracting the previous sentence as an important sentence and should be included in the summary. Finally, the binary decision is modeled by the sigmoid layer as follows:

$$p(y(t) = 1|D) = \sigma(MLP(\bar{h}_t : h_t)) \quad (2)$$

MLP is a multi-layer neural network.

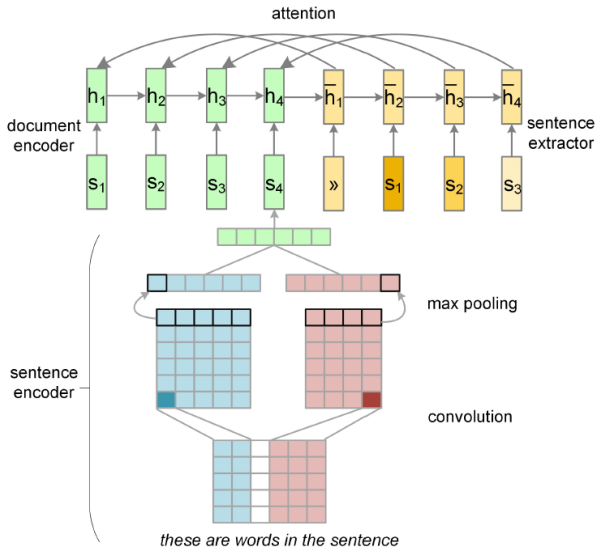


Figure 1. Encoder and sentence extractor of NeuralSum [5]

**Pointer Generator Network (PGN)** [7] is a neural network-based method developed to perform abstractive summarization. The method combines the pointer generator into a sequence-to-sequence with an attention (seq2seq-attention) model. The seq2seq-attention model itself was initially developed by [22] with three components, i.e.:

- Encoder: contains a bidirectional LSTM layer to extract information from the original text. At each step, this layer reads the words one at a time and produces a hidden state sequence.
- Decoder: contains a single-layer unidirectional LSTM to create a summary after reading the original text. This layer uses information from the encoder and previous words to get the probability distribution of the next word.
- Attention mechanism: produces an intermediate hidden state in the encoder. The decoder can access this information to decide the next word.

However, the seq2seq-attention method cannot produce accurate information and more repetitions in the summary results. Therefore, they propose a pointing mechanism to solve this problem by making a soft switch, referring to the original text, or producing new words from the existing vocabulary. It is shown as generation probability ($p_{gen}$) in Figure 2.

The probabilities value between 0 and 1 are used as weights and combine vocabulary distribution ($P_{vocab}$, to generate word) with attention distribution (a, to pointing words in the source text) into the final distribution ($P_{final}$) ), as shown in the following equation:

$$p_{final}(w) = p_{gen}p_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i=w} a_i \quad (3)$$

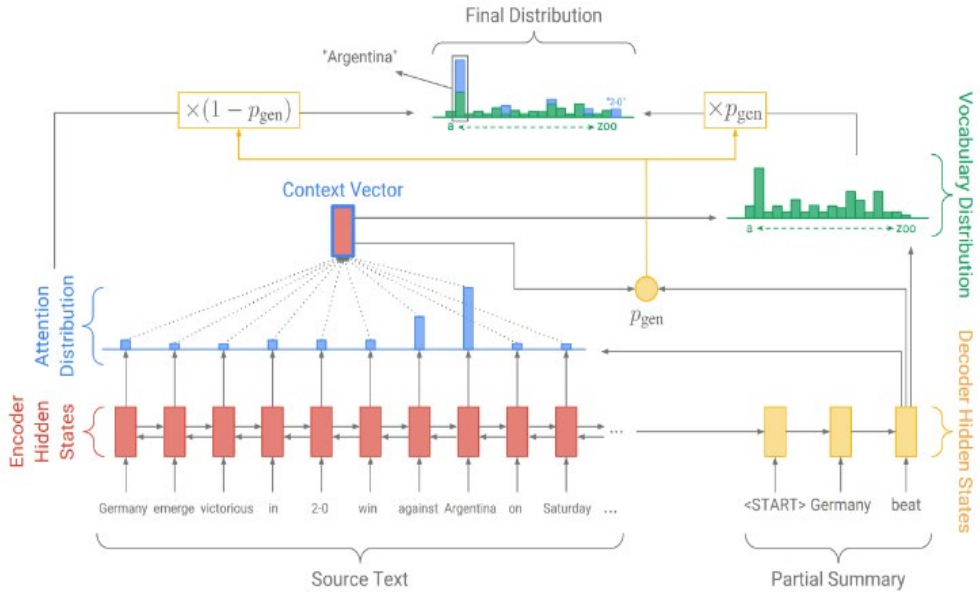


Figure 2. The architecture of Pointer Generator Network [7]

If w is an out-of-vocabulary (OOV) word, then $P_{vocab}$ will be zero, so it will refer to the original text. Otherwise, if w is not in the original text, then $\sum_{i:w_i=w} a_i$ is zero. Finally, the model will generate a new word by concatenating the output of state in the decoder's RNN model and context vector to get the probabilities of the words in the vocabulary.

**BertSum** [6] is a text summarization method that employs a pre-trained model to summarize the text extractively or abstractively. 'It uses BERT's pre-trained model [14], a model that is now being state-of-the-art in many NLP tasks, to get sentence representations in the encoder. In this case, the input of the BERT model is modified to accommodate multiple sentences, as shown in Figure 3

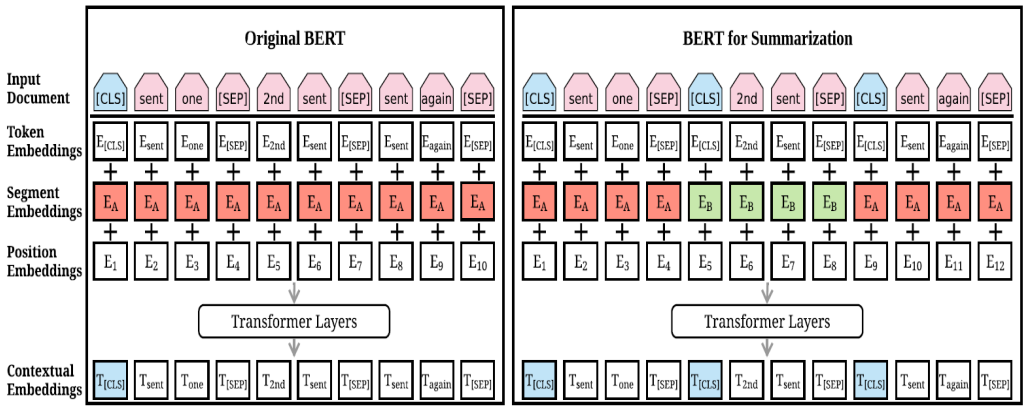


Figure 3. The architecture of the original BERT model (left) and BertSum (right) [6]

The figure above shows that the input structure of BertSum is slightly different compared to the original BERT. BertSum inserts a [CLS] token at the beginning of each sentence in the input document to collect the sentence features. Besides, BertSum uses segment embedding to distinguish multiple sentences within a document. The $E_A$ symbol marks sentences in odd segments, while $E_B$ marks sentences in even segments. For example, a document consisting of

5 sentences will have segment embeddings [$E_A$, $E_B$, $E_A$, $E_B$, $E_A$]. It is intended that document representation can be learned hierarchically, where the lower layers represent adjacent sentences while higher layers represent multi-sentence discourse. The sum vector of three embeddings (token embeddings, segment embeddings, and position embeddings) are used as input embeddings to several Transformer layers to produce contextual vectors for each token.

The extractive summarization model is built on top of the encoder by stacking several Transformer layers to get features at the document level. The output layer uses the sigmoid function, while the loss of the model uses a binary classification entropy between the predicted results and its ground truth. In predicting the summary, the model first gives a score in each sentence and then ranks from the highest to the lowest. Then the model will take the top 3 rankings as a summary sentence. During this sentence selection, the Trigram Blocking method is used to reduce redundancy. The method will ignore the candidate sentences if three words appear together (trigram overlapping) in the candidate sentences and candidate summaries.

In abstractive summarization, BertSum uses an encoder-decoder approach. The encoder uses the Bertsum pre-training model, while the decoder must be trained from the start using the six-layer Transformer. Because this difference can lead to overfitting or underfitting, the learning rates used in both are also different. In this case, the encoder learning rate is much lower than the decoder, so when the decoder becomes stable, the encoder can be trained with a more accurate gradient value. The Beam Search method is used in the decoding process until the final token of the sequence or trigram repetition is blocked.

## 3. Experiment and Evaluation

*A. Experiment*

This study investigated BertSum, Pointer Generator Network, and NeuralSum, trained on the Indonesian news articles. NeuralSum and BertSum-Ex are extractive methods, while Pointer Generator Network and BertSum-Abs are abstractive methods. Since BertSum can summarize the text in extractive and abstractive ways, we differentiate the name as BertSum-Ex and BertSum-Abs. We used IndoSum This research uses a dataset from IndoSum, which introduced by Kurniawan and Louvan[4] as a dataset in this experiment, So far, to Thwhich initially contained 18,774 articles and is consist of 5 folds. However, in this study, we compiled the data into a single fold. We then split it into a new ratio by 70% training, 10% validation, and 20% testing data, as shown in Table 1.

Table 1. Corpus Statistic

| | Train | Valid | Test | Total |
|---|---|---|---|---|
| #Article | 13517 | 1502 | 3755 | 18774 |
| Avg # of words/sent | 19.78 | 19.55 | 19.69 | 19.75 |
| Avg # of words/article | 320.41 | 318.07 | 321.57 | 320.45 |
| Avg # of sents/article | 17.02 | 17.09 | 17.19 | 17.06 |
| Avg # of words/ext. summary | 70.56 | 70.42 | 70.73 | 70.58 |
| Avg # of sents/ext. summary | 3.40 | 3.41 | 3.42 | 3.41 |
| Avg # of words/abs. summary | 67.99 | 68.01 | 68.03 | 68.00 |
| Avg # of sents/abs. summary | 3.48 | 3.48 | 3.49 | 3.48 |
| Max # of words/sent | 88 | 53 | 68 | 88 |
| Max # of words/article | 1439 | 1015 | 1388 | 1439 |
| Max # of sents/article | 76 | 66 | 67 | 76 |
| Max # of words/ext. summary | 196 | 132 | 214 | 214 |
| Max # of sents/ext. summary | 12 | 12 | 11 | 12 |
| Max # of words/abs. summary | 121 | 105 | 112 | 121 |
| Max # of sents/abs. summary | 11 | 8 | 9 | 11 |

Based on the table, it shows that the dataset have equally document length and summary length so that the training data, validation data and testing data also have an equal distribution.

*C. Evaluation*

We applied ROUGE (Rouge-1, Rouge-2, and Rouge-L) to the summary results as an evaluation metric Then wof ingeach summarization method as well as observations to its summary. ROUGE (Recall-Oriented Understanding for Gisting Evaluation) is a method for measuring the summarization results that works by comparing the summary results generated by a computer (called a candidate summary) with an ideal summary made by humans (called a reference summary) [15]. ROUGE-N is the number of n-grams that overlap in the candidate summary and reference summary. Meanwhile, ROUGE-L is the longest common Subsequence (LCS) ratio that appears together in the candidate summary and reference summary.

Each ROUGE metric has Recall, Precision, and F1-score values. Given $w_{overlapped}$ as the number of overlapping words in candidate summary and reference summary, $w_{ref}$ is the number of words in reference summary, and $w_{cand}$ is the number of words in candidate summary, the following equations can calculate the metric:

$$Recall = \frac{w_{overlapped}}{w_{ref}} \tag{4}$$

$$Precision = \frac{w_{overlapped}}{w_{cand}} \tag{5}$$

$$F1score = \frac{2 \times Recall \times Precision}{(Recall + Precision)} \tag{6}$$

In this study, the words refer to the unigram for ROUGE-1, bigram for ROUGE-2, and LCS for ROUGE-L. We used the abstractive summary of IndoSum as a reference summary to calculate the ROUGE score.

The experiment starts with data preparation because each method requires a different input format. While BertSum accepts data in JSON format with tokenized words and sentences, PGN needs data in binary format, which is separated into several chunks. Then, we perform some pre-processing

*D. Experiment Results*

The implementation of NeuralSum is refer to [15] because they already used this method to evaluate the Indosum dataset. However, in this study we preprocess the article by removing the online news name appearing at the beginning of the article, remove the author's initials, case folding, and removing quote marks because it will interfere with the training process.

In this study, we used hyperparameter by following the reference paper of each method, except BertSum. However, it needs some adjustments due to limited computing resources. For example, NeuralSum set up word embedding dimensions of 50, and the maximum length of the document was 15 sentences. Each sentence has a maximum of 50 words, and the summary contains at most three sentences. Meanwhile, PGN is run with word embedding dimensions of 128, and each document has a maximum of 400 words. The summary includes a maximum of 100 words during training and 120 words during testing.

BertSum set up a document length of 5-150 sentences, and each sentence contains 5-200 tokens. In each summary sentence, we allow 20-121 tokens. The training steps ere adjusted to 5000 on the extractive method, while in the abstractive method, we changed to 6000. This training steps parameter is modified from the original paper, which trained the system at 50000 steps in extractive and 200000 steps in the abstractive method. We need to be careful in determining these steps since it will affect the learning rate used in this method, as shown in the following equations:

$$lr_E = 2e^{-3}\min\left(step^{-0.5}, step.warmup_E^{-1.5}\right) \quad (7)$$

$$lr_D = 0.1\ \min\left(step^{-0.5}, step.warmup_D^{-1.5}\right) \quad (8)$$

During BertSum training, we use some pre-trained models, such as 'bert-base-uncased', "cahya/bert-base-Indonesian-522M"[1], and "indobenchmark/indobert-base-p1"[2][23]. The last two are Indonesian pre-trained models. We have to change some reserved tokens in BertSum since they were not available in Indonesian pre-trained vocabulary, i.e. :[unused0], [unused1], [unused2]. These tokens were used to mark the beginning of sentences (BOS), the end of sentences (EOS), and the separator between sentences in the summary results. We altered the BOS token of [unused0] to [CLS], the EOS token of [unused1] to [UNK], and the separator token of [unused2] to [SEP].

The experiment results are shown in Table 2. We also report the Lead-3 method in the table since it is commonly used as a baseline in text summarization. The Lead-3 simply selects the first three sentences as the summary. In extractive summarization, BertSum trained with the Indonesian's pre-trained model generally had the best ROUGE score. Although its ROUGE-1 and ROUGE-2 score is equal to Lead-3, DNN have more parameters for fine-tuning than the Lead-3 method. lead-3 gives better score on Rouge-1 than the other extractive methods. also despite beingan English pre-ed.BertSumcould notd in abstractive summarization. It results in a lower F1-when compared to because PGN has a coverage mechanism to solve the repeated word problem.

Table 1. ROUGE Score of Testing Data

| Method | R1-R | R1-P | R1-F | R2-R | R2-P | R2-F | RL-R | RL-P | RL-F |
|---|---|---|---|---|---|---|---|---|---|
| **Extractive** | | | | | | | | | |
| Lead-3 | **0.66** | 0.70 | **0.67** | **0.59** | 0.63 | **0.60** | 0.63 | 0.67 | 0.64 |
| NeuralSum | 0.63 | **0.71** | 0.65 | 0.57 | **0.66** | **0.60** | 0.63 | **0.71** | 0.65 |
| BertSum-Ext-En | 0.65 | 0.70 | 0.66 | 0.58 | 0.62 | 0.59 | **0.65** | 0.69 | **0.66** |
| BertSum-Ext-Id | **0.66** | 0.70 | **0.67** | 0.58 | 0.62 | 0.59 | 0.65 | 0.69 | **0.66** |
| BertSum-Ext-IndoBERT | **0.66** | 0.70 | **0.67** | **0.59** | 0.62 | **0.60** | **0.66** | 0.69 | **0.66** |
| **Abstractive** | | | | | | | | | |
| PGN | 0.65 | **0.70** | **0.67** | **0.59** | **0.63** | **0.61** | **0.63** | **0.67** | **0.65** |
| BertSum-Abs-En | 0.59 | 0.66 | 0.62 | 0.50 | 0.55 | 0.52 | 0.58 | 0.65 | 0.61 |
| BertSum-Abs-Id | 0.62 | 0.63 | 0.60 | 0.53 | 0.54 | 0.51 | 0.62 | 0.62 | 0.59 |
| BertSum-Abs-IndoBERT | **0.67** | 0.49 | 0.55 | 0.54 | 0.39 | 0.45 | 0.65 | 0.47 | 0.54 |

*Pre-trained model used in the BertSum methods as follows:
BertSum-Ext/Abs-En: *bert-base-uncased*, BertSum- Ext/Abs -Id: *bert-base-Indonesian-522M*,
BertSum- Ext/Abs -IndoBERT: *indobert-base-p1*

Seeing that the Lead-3 method performed so well, we did further experiments to investigate the effect of important sentences' position in the document on summary results. We split the testing data into three categories, i.e.:

- *[3] of sents*: three of n important sentences are in lines 1-3
- *[0,2] of sents*: maximum of two important sentences are in lines 1-3
- *[0] of sents*: none of the important sentences are in lines 1-3. It is a subset of the *[2] of sents*' dataset

The number of datasets in each category is given in Table 3 is 4

[1] https://huggingface.co/cahya/bert-base-indonesian-522M
[2] https://huggingface.co/indobenchmark/indobert-base-p1

Table 2. Testing Data Based on The Position of Important Sentence

| Category | #Testing Data |
|---|---|
| [3] of sents | 1420 |
| [0,2] of sents | 2335 |
| [0] of sents | 254 |

Table 3. Rouge Score Based on The Position of Important Sentence

| **Category** | **R1-R** | **R1-P** | **R1-F** | **R2-R** | **R2-P** | **R2-F** | **RL-R** | **RL-P** | **RL-F** |
|---|---|---|---|---|---|---|---|---|---|
| **A. [3] of sents dataset** | | | | | | | | | |
| ***Extractive*** | | | | | | | | | |
| Lead-3 | **0.81** | **0.93** | **0.86** | **0.79** | **0.90** | **0.83** | **0.81** | **0.92** | **0.85** |
| NeuralSum | 0.75 | 0.90 | 0.81 | 0.72 | 0.86 | 0.77 | 0.74 | 0.89 | 0.80 |
| BertSum-Ext-En | 0.77 | 0.88 | 0.81 | 0.73 | 0.84 | 0.77 | 0.77 | 0.88 | 0.81 |
| BertSum-Ext-Id | 0.75 | 0.85 | 0.79 | 0.71 | 0.80 | 0.74 | 0.75 | 0.85 | 0.79 |
| BertSum-Ext-IndoBERT | 0.77 | 0.86 | 0.80 | 0.72 | 0.81 | 0.76 | 0.76 | 0.86 | 0.80 |
| ***Abstractive*** | | | | | | | | | |
| PGN | **0.78** | 0.82 | **0.80** | **0.74** | **0.78** | **0.75** | **0.77** | 0.82 | **0.79** |
| BertSum-Abs-En | 0.73 | 0.82 | 0.77 | 0.66 | 0.74 | 0.69 | 0.73 | 0.81 | 0.76 |
| BertSum-Abs-Id | 0.69 | **0.83** | 0.72 | 0.64 | **0.78** | 0.67 | 0.69 | **0.83** | 0.72 |
| BertSum-Abs-IndoBERT | 0.77 | 0.57 | 0.64 | 0.67 | 0.50 | 0.56 | 0.75 | 0.56 | 0.63 |
| **B. [0,2] of sents dataset** | | | | | | | | | |
| ***Extractive*** | | | | | | | | | |
| Lead-3 | 0.56 | 0.57 | 0.55 | 0.47 | 0.47 | 0.46 | 0.52 | 0.52 | 0.51 |
| NeuralSum | 0.55 | **0.63** | 0.57 | 0.48 | **0.54** | 0.49 | 0.52 | 0.59 | 0.54 |
| BertSum-Ext-En | 0.57 | 0.58 | 0.56 | 0.48 | 0.48 | 0.47 | 0.56 | 0.57 | 0.55 |
| BertSum-Ext-Id | 0.60 | 0.60 | 0.59 | 0.51 | 0.50 | 0.50 | 0.59 | 0.59 | 0.58 |
| BertSum-Ext-IndoBERT | **0.61** | 0.61 | **0.60** | **0.52** | 0.51 | **0.51** | **0.60** | **0.60** | **0.59** |
| ***Abstractive*** | | | | | | | | | |
| PGN | 0.59 | **0.61** | **0.60** | **0.50** | **0.52** | **0.50** | 0.58 | **0.60** | **0.59** |
| BertSum-Abs-En | 0.51 | 0.56 | 0.53 | 0.39 | 0.44 | 0.41 | 0.49 | 0.55 | 0.51 |
| BertSum-Abs-Id | 0.55 | 0.56 | 0.53 | 0.44 | 0.45 | 0.42 | 0.54 | 0.55 | 0.52 |
| BertSum-Abs-IndoBERT | **0.61** | 0.44 | 0.50 | 0.47 | 0.34 | 0.38 | **0.59** | 0.43 | 0.48 |
| **C. [0] of sents dataset** | | | | | | | | | |
| ***Extractive*** | | | | | | | | | |
| Lead-3 | 0.18 | 0.22 | 0.20 | 0.04 | 0.05 | 0.05 | 0.12 | 0.15 | 0.13 |
| NeuralSum | 0.29 | 0.43 | 0.34 | 0.20 | 0.30 | 0.23 | 0.25 | 0.37 | 0.28 |
| BertSum-Ext-En | 0.36 | 0.39 | 0.37 | 0.23 | 0.25 | 0.24 | 0.34 | 0.37 | 0.35 |
| BertSum-Ext-Id | 0.39 | 0.43 | 0.41 | 0.27 | 0.30 | 0.28 | 0.38 | 0.41 | 0.39 |
| BertSum-Ext-IndoBERT | **0.42** | **0.44** | **0.42** | **0.30** | **0.31** | **0.30** | **0.40** | **0.42** | **0.41** |
| ***Abstractive*** | | | | | | | | | |
| PGN | 0.33 | **0.34** | **0.33** | **0.18** | **0.18** | **0.18** | 0.31 | **0.32** | **0.31** |
| BertSum-Abs-En | 0.24 | 0.26 | 0.24 | 0.09 | 0.09 | 0.09 | 0.22 | 0.24 | 0.22 |
| BertSum-Abs-Id | 0.30 | 0.28 | 0.28 | 0.14 | 0.12 | 0.13 | 0.28 | 0.26 | 0.26 |
| BertSum-Abs-IndoBERT | **0.35** | 0.26 | 0.29 | 0.16 | 0.12 | 0.13 | **0.33** | 0.24 | 0.27 |

*D. Evaluation*

in Table 4we observed that all three methods (NeuralSum, BertSum, and PGN) are likely to select . It can be seen that in group A, ROUGE, and unsurprisingly, L outperforms all the The ROUGE scoregetsWe further compare these methods based on summary techniques, extractive and abstractive methods in the following subsections.

*1) Extractive Summarization*

Extractive summarization methods work by selecting the important sentences as summary candidates. Based on our observation, NeuralSum tends to choose the leading sentences in the document compared to BertSum-Ext, as shown in Figure 4. More than 60% of testing documents in NeuralSum get the first three sentences as a summary, but only a third of those in BertSum do. Therefore, as shown in Table 4, NeuralSum gets a lower Rouge score than BertSum-Ext, especially in groups B and C of datasets, when not all summary sentences are at the beginning of the document.

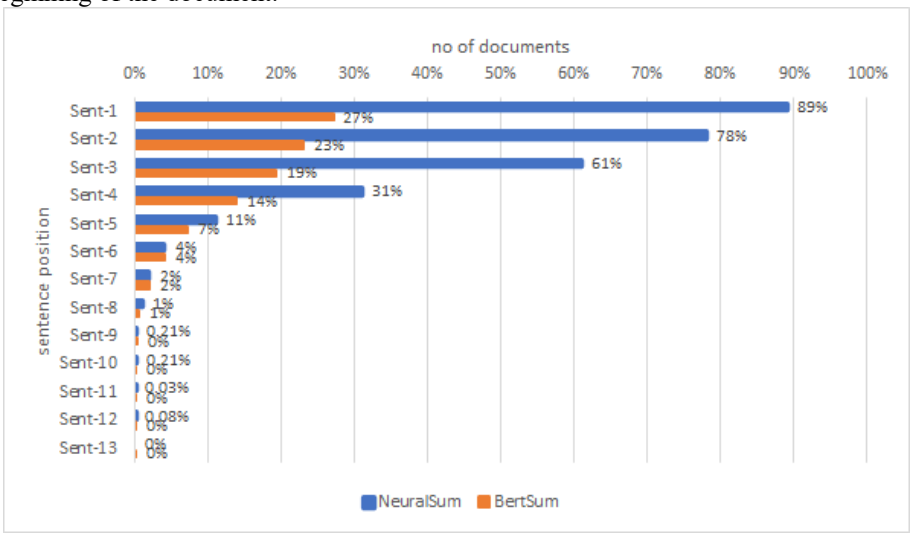


Figure 4. Extraction of important sentences in NeuralSum and BertSum-Ext based on sentence position in a document

Another finding in our experiment is that NeuralSum sometimes produces a blank summary, as shown in Fig.5 that 71 documents did not extract important sentences in the article. It may happen because the sentences labeling processes in NeuralSum and BertSum-Ext are different. NeuralSum performs sequences labeling that allows the accumulation of prediction errors from the previous sentence predictions. This error accumulation may result in no sentence being identified as important. Meanwhile, BertSum-Ext directly applies the MLP layer and sigmoid activation to the transformer encoder output to predict each sentence label. Moreover, NeuralSum doesdoes not consider other instances in the sentence selection process, whether another sentence is already in the summary candidates or not. Meanwhile, BertSum-Ext used Trigram Blocking to select important sentences that continuously check the status of the candidate summary. As a result, it will ignore the candidate sentence if three similar words appear in the candidate summary. Based on our experiment, the shortest summary produced by BertSum-Ext is 25 words, PGN is 36 words, and BertSum-Abs is 12 words.

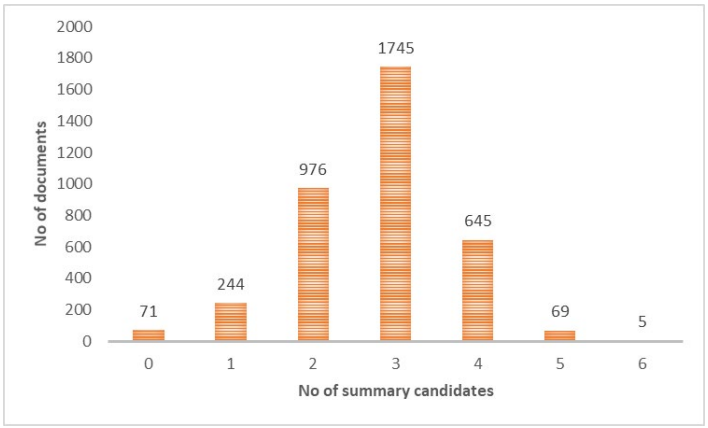


Figure 5. The number of important sentences selected in NeuralSum

Most of the summarization methodsrequire a , such as,which sin an article to be As an illustrationLikewise, can only, even though contains up to

*2) Abstractive Summarization*

As shown in Table 4, BertSum with IndoBERT pre-trained model produces low precision despite the high recall, thus lowering the F1 score. This result is opposite from the extractive method that BertSum-Ext-IndoBERT outperforms other extractive summarization methods, especially on the second and third groups of datasets. We further compare two abstractive methods, Pointer Generator Network and BertSum-Abs, by observing its summary outputs since these methods generate the summary word by word. Finally, an example of summarization results is presented in Table 5, which consists of the original article, gold summary, and generated summary of each method.

Table 4. A Sample Article and Its Summarization

| |
|---|
| **Original article** |
| bagi orang-orang yang mengalami gluten - toleran atau penderita celiac yang gemar menikmati vodka , kini sudah ada vodka bebas gluten yang dapat dinikmati tanpa khawatir .<br>produsen minuman vodka , stoli , baru saja merilis produk spesial mereka yang sebelumnya tidak pernah diproduksi .<br>vodka stolichnaya yang terbuat dari 88 persen jagung dan 12 persen gandum menjadi menu baru andalan stoli .<br>meski ada beberapa produsen kecil vodka yang telah mengeluarkan produk bebas gluten , stoli mengklaim sebagai merek global pertama yang melakukannya .<br>lembaga the beverage testing institute memberikan nilai produk bebas gluten ini 92 poin dari 100 .<br>karena semakin banyak orang yang menggabungkan pilihan minuman bebas gluten dalam ulasan diet mereka , kami pun ingin menciptakan produk yang tidak diragukan lagi untuk konsumen kami .<br>di dalam botol ini benar-benar bebas gluten , kata presiden dan ceo stoli patrick piana , dikutip daily meal .<br>kami ingin memenuhi permintaan dan memastikan bahwa resep bebas gluten kami benar-benar dibuat dengan bahan-bahan yang juga bebas gluten .<br>tapi kami juga tetap mempertahankan karakter kami yang berani namun tetap disajikan dengan halus .<br>produk stoli ini diklaim dibuat secara alami bebas gluten dengan jagung dan gandum yang tumbuh di daerah tambov , rusia . |
| **Gold summary** |
| terdapat vodka bebas gluten yang aman untuk penderita celiac atau gluten - toleran . produsen minuman vodka , stoli , baru saja merilis produk spesial yang sebelumnya belum pernah diproduksi . vodka stolichnaya terbuat dari 88 persen jagung dan 12 persen gandum menjadi menu baru andalan stoli. lembaga the beverage testing institute memberikan nilai produk bebas gluten ini 92 poin dari 100 . |

| |
|---|
| **Pointer Generator Network** |
| bagi orang-orang yang mengalami gluten - toleran atau penderita celiac yang gemar menikmati vodka , kini sudah ada vodka bebas gluten yang dapat dinikmati tanpa khawatir . produsen minuman vodka , stoli , baru saja merilis produk spesial mereka yang sebelumnya tidak pernah diproduksi . meski diketahui presiden , trump akan memperketat perbatasan as - meksiko dengan membangun tembok tinggi . |
| **BertSum-Abs-En** |
| bagi orang-orang yang mengalami gluten - toleran atauderita celiac yang gemar menikmati sudah ada suv bebas gluten yang dapat dinikmati tanpat dinikmati tanpa khawatir .produsen minuman ellen , stoli , baru saja merilis produk spesial mereka yang sebelumnya tidak pernah diproduksi .di dalam botol ini pun ingin menciptakan produk yang tidak diragukan lagi untuk konsumen kami . |
| **BertSum-Abs-Id** |
| bagi bagi yang mengalami gluten - rentan atau penderita celeron yang gemar menikmati vodka , kini sudah ada vodka bebas gluten yang dapat dinikmati tanpa khawatir .produsen minuman vodka , svi , baru saja merilis produk spesial mereka yang sebelumnya tidak pernah diproduksi .namun ada beberapa produsen kecil vodka yang telah mengeluarkan produk bebas gluten , sven mengklaim sebagai merek bebas gluten .lembaga the beverage institute memberikan nilai produk produk bebas glut gluten ini .yayasan oleh |
| **BertSum-Abs-IndoBERT** |
| bagi orang yang mengalami alkohol yang gemar menikmati vodka bebas menikmati vodka , kini sudah ada vodka dapat dinikmati tanpa khawatir .produsen minuman vodkhnaya yang telah mengeluarkan produk spesial mereka yang sebelumnya tidak pernah diproduksi .vodka stoli .vodkhnaya yang terbuat dari 88 persen jagung dan 12 persen beras menjadi menu baru andalan stoli . lembaga the beverage institute memberikan nilai produk bebas saji saji . . untuk membuat produk indonesia pertama yang dilakukan . untuk mengeluarkan produk bebas karbohidrat , stoli , st alve . dari 99 persen jagung . dengan total jagung dan 11 persen jagung , stoli yang dilakukan karena ada beberapa produsen global pertama yang membuat produk bebasolik stoli . |

*Underlined sentences are the extractive summary, a gold summary is an abstractive summary

Based on our experiment, we found several errors highlighted in the abstractive summary result as follows:

1. There are many words/phrases repetitions in summary produced by Bertsum-Abs. For example, in Table 5, we can find this kind of error in Bertsum-Abs-IndoBert's summary, e.g., "…bebas saji saji." (*free service service*). It happens due to no coverage mechanism in BertSum, while trigram blocking is used only to reduce repeating sentences instead of words.
2. The syntactic errors have resulted in sentences not following grammatical rules and cannot be adequately understood. This error also can be found in the summary produced by Bertsum-Abs-IndoBert: "untuk membuat produk indonesia pertama yang dilakukan." (*to make the first Indonesian product.*). Since this sentence is incomplete, it makes us difficult to understand.
3. The semantic errors were found because the summary is have out of context. It is mostly found in summary produced by Pointer Generator Network. For example, the last phrase "meski diketahui presiden , trump akan memperketat perbatasan as - meksiko dengan membangun tembok tinggi ." (*although it is known by the president, trump will tighten the us -mexico border by building a high wall.*) is out of context from the beverage topic and is not written in the original article.

Meaningless out-of-vocabulary (OOV) found in Bertsum-Abs, such as "atauderita", "sven", "bebasolik", etc. It happens since BertSum creates new words based on vocabulary (sub-words) produced by Bert's pre-trained model. Surprisingly, these errors still occur even though the model applies the Indonesian pre-trained model.

Furthermore, it seems obvious that the ROUGE score does not support the synonym at the lexical level. So many synonym words are considered different, e.g., , that it leads to lowers the ROUGE score. Therefore, evaluation open problem Finally, we present a brief comparison of

each method based on our experiment, both in architecture and summary results, as shown in Table 6.

Table 5. Methods Comparison

| Comparison Item | NeuralSum | BertSum-Ext | PGN | BertSum-Abs |
|---|---|---|---|---|
| Sent./doc. representation | CNN + LSTM | BERT[9] | Bi-LSTM + attention mechanism | BERT[9] |
| Sent. selection/ word generation | Sent. scoring with LSTM + sigmoid and ranking the score | Sent. scoring with Transformer and ranking with Trigram blocking | LSTM + beam search +pointer switch + coverage mechanism | Transformer + beam search + trigram blocking |
| Use pre-trained model | no | BERT pre-trained model | no | BERT pre-trained model |
| Maximum input length | Document length, defined by the user | Max 512 tokens (length of BERT's positional embedding) | Vocab size, defined by the user | Max 512 (length of BERT's positional embedding) |
| Tendency to select leading sentences as summary | yes | yes | no | no |
| Produces blank summary | yes | no | no | no |
| Repeated word/phrase | no | no | no | yes |
| Syntactic error | no | no | yes | yes |
| Semantic error (out of context) | no | no | yes | yes |
| OOV | no | no | no | yes |

## 4. Conclusion

This study investigated BertSum and Pointer Generator Network (PGN) for Indonesian text summarization. We used these methods as a comparison to NeuralSum, which was well performed on the IndoSum corpus. While Neuralsum only summarized extractively and PGN did abstractively, BertSum was proposed to summarize using both techniques. However, the three methods used an encoder-decoder architecture and required high computational resources, so we adjusted some parameters.

In our experiment, we used the IndoSum corpus by managing it differently from the original paper. Therefore, we compiled the dataset into a single fold. We split it into a new ratio for training, validation, and testing dataset. Finally, we compare the ROUGE score resulting from the experiment to its abstractive summary in the evaluation process since it is the gold summary of the IndoSum dataset.

In extractive summarization, BertSum with the Indonesian pre-trained model performed well compared to NeuralSum. However, it still gave poor results when generating an abstractive summary. It produced a lot of meaningless OOV and repetitive words leading to a lower ROUGE score than PGN. It may be a problem when generating sub-words by Bert Tokenizer. A coverage mechanism by PGN is known to be able to avoid this word repetition. However, PGN still has pointer switching issues, so it sometimes produces summaries out of context. Meanwhile, NeuralSum is modest compared to the other two methods, so the training process is also faster. Yet, this method can produce a blank summary when there is no sentence classified as important information.

**5. Acknowledgment**
This research was partially supported by the Ministry of Research and Technology/National Research and Innovation Agency, the Republic of Indonesia, under the SAINTEK 2018 project and Doctoral Dissertation Research Grant 2020.

**6. References**
[1] I. Mani and M. T. Maybury, Advances in automatic text summarization, vol. 26, no. 2. MIT Press, 1999.
[2] H. P. Luhn, "The Automatic Creation of Literature Abstracts," IBM J. Res. Dev., vol. 2, no. 2, pp. 159–165, 1958.
[3] M. Kågebäck, O. Mogren, N. Tahmasebi, and D. Dubhashi, "Extractive Summarization using Continuous Vector Space Models," in Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC), 2014, pp. 31–39.
[4] K. Kurniawan and S. Louvan, "IndoSum: A New Benchmark Dataset for Indonesian Text Summarization," in The International Conference on Asian Language Processing (IALP 2018), 2018.
[5] J. Cheng and M. Lapata, "Neural Summarization by Extracting Sentences and Words," in The 54th Annual Meeting of the Association for Computational Linguistics, 2016, pp. 484–494.
[6] Y. Liu and M. Lapata, "Text Summarization with Pretrained Encoders," in The 2019 Conference on Empirical Methods in Natural Language Processing and The 9th International Joint Conference on Natural Language Processing, 2019, pp. 3728–3738.
[7] A. See, P. J. Liu, and C. D. Manning, "Get To The Point: Summarization with Pointer-Generator Networks," in The 55th Annual Meeting of the Association for Computational Linguistics, 2017, pp. 1073–1083.
[8] R. Chandraseta and M. L. Khodra, "Composing Indonesian Paragraph for Biography Domain using Extractive Summarization," in 2019 International Conference on Advanced Informatics: Concepts, Theory, and Applications (ICAICTA 2019), 2019, pp. 1–5.
[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in NAACL-HLT 2019, 2019, pp. 4171–4186.
[10] F. Koto, "A publicly available Indonesian corpora for automatic abstractive and extractive chat summarization," in Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016, 2016, pp. 801–805.
[11] G. S. Budhi, R. Intan, and S. R. R, "Indonesian Automated Text Summarization," in International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIT), 2007.
[12] Y. Herdiyeni, A. Ridha, and J. Adisantoso, "Text Feature Weighting For Summarization Of Document Bahasa Indonesia Using Genetic Algorithm," Int. J. Comput. Sci. Issues, vol. 9, no. 3, pp. 1–6, 2012.
[13] Silvia, P. Rukmana, V. R. Aprilia, D. Suhartono, R. Wongso, and Meiliana, "Summarizing Text for Indonesian Language by Using Latent Dirichlet Allocation and Genetic Algorithm," Int. Conf. Electr. Eng. Comput. Sci. Informatics, vol. 1, no. August, pp. 148–153, 2014.
[14] P. P. Tardan, A. Erwin, K. I. Eng, and W. Muliady, "Automatic Text Summarization Based on Semantic Analysis Approach for Documents in Indonesian Language," in 2013 International Conference on Information Technology and Electrical Engineering (ICITEE 2013), 2013, pp. 47–52.
[15] D. Gunawan, A. Pasaribu, R. F. Rahmat, and R. Budiarto, "Automatic Text Summarization for Indonesian Language Using TextTeaser," in IOP Conference Series: Materials Science and Engineering, 2017.
[16] R. B. Hutama, A. R. Barakbah, and A. Helen, "Indonesian news auto summarization in infrastructure development topic using 5W+1H consideration," in Proceedings -

International Electronics Symposium on Knowledge Creation and Intelligent Computing, IES-KCIC 2017, 2017, vol. 2017-Janua, pp. 258–264.
[17] P. M. Sabuna and D. B. Setyohadi, "Summarizing Indonesian text automatically by using sentence scoring and decision tree," in 2017 2nd International Conferences on Information Technology, Information Systems and Electrical Engineering ( ICITISEE 2017), 2017, pp. 1–6.
[18] F. Maryana, A. Kurniawati, and D. Agusten, "Term frequency method for automated text summarization application of Indonesian news article," in The 3rd International Conference on Informatics and Computing (ICIC 2018), 2018, pp. 1–7.
[19] Y. Dong, "A Survey on Neural Network-Based Summarization Methods." pp. 1–16, 2018.
[20] K. Halim, H. Novianus Palit, and A. N. Tjondrowiguno, "Penerapan Recurrent Neural Network untuk Pembuatan Ringkasan Ekstraktif Otomatis pada Berita Berbahasa Indonesia," J. Infra, vol. 8, no. 1, pp. 221–227, 2020.
[21] Z. Cai, N. Lin, C. Ma, and S. Jiang, "Indonesian Automatic Text Summarization Based on A New Clustering Method in Sentence Level," in International Conference on Big Data Engineering (BDE 2019), 2019, pp. 30–35.
[22] R. Nallapati, F. Zhai, and B. Zhou, "SummaRuNNer : A Recurrent Neural Network based Sequence Model for," in The Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 3075–3081.
[23] B. Wilie et al., "IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding," in The 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, 2020.

**Rini Wijayanti** is currently a doctoral student at the School of Electrical Engineering and Informatics, Institute of Technology Bandung (ITB). Her research interests include natural language processing, machine learning, deep learning, and artificial intelligence.

**Masayu L. Khodra** graduated the bachelor in Informatics from Institute of Technology Bandung (ITB) in 1999. She also received master and doctoral degrees in Informatics from ITB in 2006 and 2012, respectively. She is currently a faculty member in the School of Electrical Engineering and Informatics at Institute of Technology Bandung. Her research interests include text classification, text summarization, expert system, machine learning and artificial intelligence.

**Dwi H. Widyantoro** received the MS and Ph.D. degrees in computer science from Texas A&M University in 1999 and 2003, respectively. He is currently a faculty member in the School of Electrical Engineering and Informatics at Institute of Technology Bandung. His research interests include machine learning, deep learning, information summarization, information extraction, information classification as well as pattern recognition.