Transformer-based abstractive indonesian text summarization

Miracle Aurelia, Sheila Monica, Abba Suganda Girsang
Department of Computer Science, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia

Article Info

Article history:

Received Jan 26, 2024
Revised May 14, 2024
Accepted Jun 18, 2024

Keywords:

Abstractive summarization
BART model
ChatGPT augmentation
IndoSum
Liputan6

ABSTRACT

The volume of data created, captured, copied, and consumed worldwide has increased from 2 zettabytes in 2010 to over 97 zettabytes in 2020, with an estimation of 181 zettabytes in 2025. Automatic text summarization (ATS) will ease giving points of information and will increase efficiency at the time consumed to understand the information. Therefore, improving ATS performance in summarizing news articles is the goal of this paper. This work will fine-tune the BART model using IndoSum, Liputan6, and Liputan6 augmented dataset for abstractive summarization. Data augmentation for Liputan6 will be augmented with the ChatGPT method. This work will also use r ecall-oriented understudy of gisting evaluation (ROUGE) as an evaluation metric. The data augmentation with ChatGPT used 10% of the clean news article from the Liputan6 training dataset and ChatGPT generated the abstractive summary based on that input, culminating in over 36 thousand data for the model's fine-tuning. BART model that was finetuned using Indosum, Liputan6, and augmented Liputan6 dataset has the best ROUGE-2 score, outperforming ORACLE's model although ORACLE still has the best ROUGE-1 and ROUGE-L score. This concludes that fine-tuning the BART model with multiple datasets will increase the performance of the model to do abstractive summarization tasks.

Corresponding Author:

Miracle Aurelia
Department of Computer Science, BINUS Graduate Program - Master of Computer Science
Bina Nusantara University
Jakarta, Indonesia
Email: miracle.aurelia@binus.ac.id

1. INTRODUCTION

Nowadays, the volume of data created, captured, copied, and consumed worldwide has increased from 2 zettabytes in 2010 to over 97 zettabytes in 2020, with an estimation of 181 zettabytes in 2025 [1]. Automatic text summarization (ATS) can help reduce time spent on understanding information by simplifying the distribution of points and using extractive and abstractive approaches to extract words from the original text [2]. ATS models include neural networks, convolutional neural networks, recurrent neural networks, and transformers. Recurrent neural networks are suitable for NLP tasks but struggle with input length due to sequential processing. Transformer solves this issue by relying on an attention mechanism. The Transformer model includes attention layers, feed-forward networks, and a normalization layer. Attention is computed using dot-product multiplication for decoding and self-attention [3], [4]. Facebook AI introduced BART, a transformer model that combines bidirectional and auto-regressive transformers. Previous research [5]-[7] used BART for abstractive summarization, outperform previous work on recall-oriented understudy of gisting evaluation (ROUGE) metrics. Research on Indonesian datasets, including LEAD-n, Oracle, PTGen, PTGen+Cov, mBERT, and IndoBERT, shows Oracle has the highest overall

ROUGE score [8]. Data fine-tuning is a technique that adjusts input data based on pre-trained models, increasing the performance of black box models [9]. BART is effective for fine-tuned text generation and has good results when used for abstractive summarization for the XSum dataset [7]. Multiple data fine tuning is training a single model on several upstream and downstream tasks that performs better than standard transfer learning. Specifically, multiple dataset fine tuning has a regularizing effect and helps the network learn feature representations that transfer between different datasets, which is why it dramatically increases performance on rare classes in downstream job [10]. One of the researchers fine-tuned BART-large-CNN with SAMSum dataset that showed enhancement on the ROUGE score for the fine-tuned model [11].

Data augmentation is another approach for further boosting performance that enhances performance by training a pre-trained model on new data, reducing brute-force memorization and creating regularization effects [12]. Data augmentation is an artificial process to generate new text by applying transformation [13]. There are some different kinds of approaches for data augmentation in word-level data augmentation and sentence-level. In word-level, there are random swap, random deletion, and synonym augmentation. For sentence level back translation and generative data augmentation is used. ChatGPT also has been used for data augmentation [14]. The previous research [14] used ChatGPT for data augmentation in contrastive learning (CCL). The result is an improvement on the ROUGE score, the effectiveness and the generalization capabilities of the proposed CCL approach, which performs better than most existing AIG approaches under natural language generation. This integration proves beneficial in mitigating data bias issues and offering superior positive and negative samples, thereby enhancing the quality of the contrastive learning process [14].

This study will fine-tune the BART model using IndoSum, Liputan6, and Liputan6 augmented dataset for abstractive summarization. The Liputan6 dataset is more suitable for abstractive summarization because Indosum is less abstractive than Liputan6 [15]. Based on that statement, this work only augments the Liputan6 dataset. The purpose of fine-tuning using multiple datasets is to train the model until it has better performance in capturing language patterns and context information so it will be useful for text summarization tasks [11]. Data augmentation for Liputan6 will be augmented with the ChatGPT method. This type of approach in fine-tuning multiple Indonesian datasets and augmenting the Indonesian dataset using the ChatGPT method will be the novelty of this work. This work will also use ROUGE for evaluation [16].

2. METHOD

In this chapter, we detail the methodology in a structured manner, beginning with an overview of the concept, followed by the steps for data preprocessing and tokenization. We then discuss the implementation of data augmentation, describe the architecture of the BART model and its training process, and conclude with an evaluation of the model using the ROUGE score.

2.1. The concept of methodology

The proposed method consists of three times tuning of the model with three datasets. The datasets used are Indosum, Liputan6, and Liputan6 which have been augmented. The first step of the fine-tuning process performs data preprocessing and tokenization for each model. The first model uses the Indosum dataset, the second model uses the Liputan6 dataset, and the third model uses the Liputan6 augmented dataset. Second step, each model will be trained using the BART model then the third step, the model will be evaluated using ROUGE score. Lastly, the fine-tuned model will be saved for the downstream task which is text summarization. The concept of methodology is shown in Figure 1.

Figure 1 explains the methodology used to build the text summarization model in this study. In the process of building and fine-tuning the abstractive Indonesian text summarization model, a novel approach is proposed by fine-tuning the BART pre-trained model with multiple datasets, incorporating diverse sources of data to improve adaptability, and understanding. Additionally, data augmentation is also implemented using the OpenAI ChatGPT API. The primary goal is to enrich the training dataset by generating abstractive summaries for Liputan6 news articles, thereby enhancing the model’s ability to produce more diverse and contextually relevant summaries.

The training pipeline begins with model 1 that will be fine-tuning BART base model on the Indosum dataset, a dataset known for providing less abstractive, more extractive summaries which can be accessed publicly. All the data in Indosum, consisting of 14,262 training data, 750 validation data, and 3,762 testing data are used in building the model. This step serves as a foundational training phase to adapt the model to the summarization task, particularly focusing on the characteristics of the Indosum dataset. The fine-tuned model is then saved locally to serve as a pre-trained model for the subsequent fine-tuning.

The pre-trained model 1 is loaded and then fine-tuned on the Liputan6 dataset, which introduces a significantly larger and more diverse set of document-summary pairs that can also be accessed publicly.

This step is also shown as model 2 in figure 1. It has 193,883 training data and 10,972 testing and validation data. In this study, only around 10% of the training dataset is used due to computational limitation. After the model is trained with Liputan6 dataset, data augmentation is applied, and the augmented data is trained and saved as model 3. Augmentation is done not only to introduce novelty but is also substantiated by empirical evidence from related works, showing its efficacy in improving overall model's performance. The innovative data augmentation method involves batching the Liputan6 dataset files and employing the ChatGPT API to generate abstractive summaries for each batch. The batching process is designed to efficiently process multiple files at once, enhancing the throughput of the augmentation strategy.

The clean news article from approximately 10% of Liputan6 training dataset is sent as input to the ChatGPT API, and then ChatGPT API will generate the abstractive summaries based on the input and dump the output into separate JSON files for each news article. The generated files are then augmented into the original 10% of Liputan6 training data, culminating in a combined dataset of 36,129 instances. The augmented data, comprising the original text and the generated abstractive summaries, is then used to fine-tune the model once more. This integration of ChatGPT-generated data into the fine-tuning process aims to expose the model to a more diverse range of summarization examples.

The methodology combines multiple datasets fine-tuning strategies and a novel data augmentation approach using state-of-the-art language models. Using multiple datasets allows the model to study a wide range of words and phrases, expanding its vocabulary. Through extensive exposure to diverse sentence structures in a large dataset, the model learns the language's syntactic and semantic nuances, associating words with their contextual meanings and usage patterns.

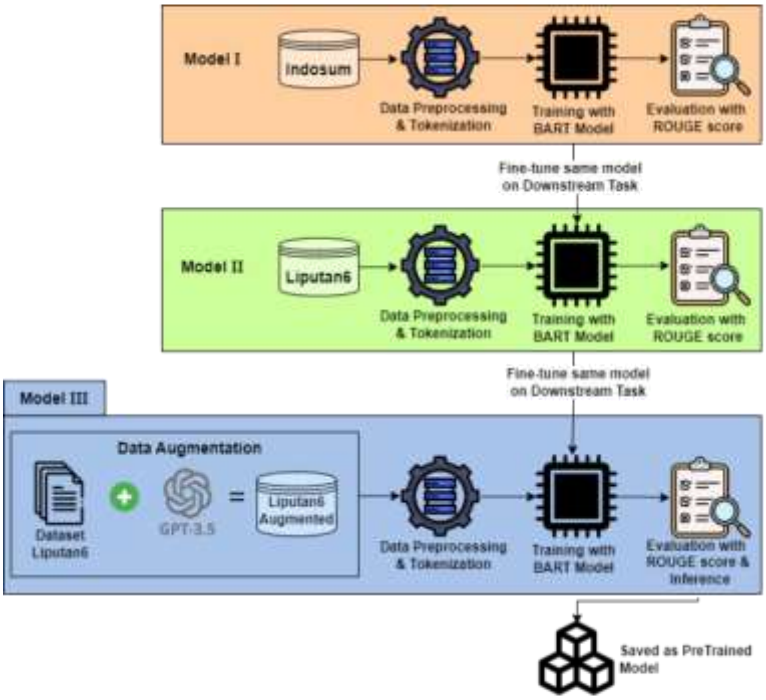

Model I
Indosum
Data Preprocessing & Tokenization
Training with BART Model
Evaluation with ROUGE score
Fine-tune same model on Downstream Task
Model II
Liputan6
Data Preprocessing & Tokenization
Training with BART Model
Evaluation with ROUGE score
Fine-tune same model on Downstream Task
Model III
Data Augmentation
Dataset Liputan6
GPT-3.5
=
Liputan6 Augmented
Data Preprocessing & Tokenization
Training with BART Model
Evaluation with ROUGE score & Inference
Saved as PreTrained Model

Figure 1. The concept of methodology

2.2. Data preprocessing and tokenization

First step is to preprocess and tokenize all data that was used as training and testing data for all the models built in this study, by flattening the original nested list structure where each sentence was represented as a list of words, and then concatenating the words in each sentence. Irrelevant features to the text summarization task like article ID, URL, and the extractive summary were also dropped to simplify the data. The data was then tokenized using the model's tokenizer to transform continuous text into numerical representation, allowing the model to learn to comprehend and process linguistic information. Table 1 shows the data preprocessing results.

The tokenized data were then given to the model's trainer to fine-tune the BART model. However, the observed limitations in the model's performance, particularly in generating sufficiently abstractive summaries for the Liputan6 dataset, prompted the introduction of a novel data augmentation strategy.

Table 1. Data preprocessing

Raw data Liputan6	Data Liputan6 after pre-processing
{“id”: 42933, “url”: “https://www.liputan6.com/news/read/42933/pekerja-seks-komersial-di-surabaya-berunjuk-rasa”, “clean_article”: [[“Liputan6”, “.”, “com”, “,”, “Surabaya”, “:”, “Sekitar”, “150”, “pekerja”, “seks”, “komersial”, “(“, “PSK”, “)”, “dari”, “lima”, “lokalisasi”, “yang”, “ada”, “di”, “Jawa”, “Timur”, “,”, “Rabu”, “(“9/10”)”, “siang”, “berunjuk”, “rasa”, “di”, “Gedung”, “DPRD”, “Surabaya”, “.”], [“Mereka”, “memprotes”, “Surat”, “Keputusan”, “Walikota”, “mengenai”, “penutupan”, “lokalisasi”, “selama”, “bulan”, “Ramadan”, “mendatang”, “.”], [“Bahkan”, “surat”, “tersebut”, “sudah”, “diserahkan”, “ke”, “DPRD”, “Surabaya”, “.”], [“Para”, “demonstran”, “yang”, “datang”, “dengan”, “dandanan”, “menor”, “itu”, “berasal”, “dari”, “lima”, “lokalisasi”, “,”, “yakni”, “Doli”, “,”, “Jarak”, “,”, “Moroseneng”, “,”, “Tambak”, “Asri”, “,”, “dan”, “Bangun”, “Rejo”, “.”], [“Selain”, “para”, “PSK”, “,”, “beberapa”, “pedagang”, “yang”, “menggantungkan”, “hidup”, “di”, “daerah”, “lokalisasi”, “juga”, “turut”, “berunjuk”, “rasa”, “.”], [“Dalam”, “orasinya”, “,”, “mereka”, “meminta”, “SK”, “Walikota”, “ditinjau”, “ulang”, “.”], [“Menurut”, “mereka”, “,”, “selama”, “bulan”, “puasa”, “mereka”, “harus”, “mengumpulkan”, “uang”, “buat”, “keluarganya”, “untuk”, “keperluan”, “di”, “Hari”, “Raya”, “Idul”, “Fitri”, “.”], [“Jika”, “nantinya”, “SK”, “tersebut”, “tetap”, “diberlakukan”, “,”, “mereka”, “berharap”, “instansi”, “terkait”, “konsisten”, “menindak”, “tegas”, “prostitusi”, “kelas”, “atas”, “yang”, “beroperasi”, “di”, “hotel-hotel”, “.”, “(“, “PIN/Benny”, “Christian”, “dan”, “Bambang”, “Ronggo”, “)”, “.”]], “clean_summary”: [[“Sekitar”, “150”, “PSK”, “di”, “Surabaya”, “,”, “Jatim”, “,”, “memprotes”, “SK”, “Walikota”, “mengenai”, “penutupan”, “lokalisasi”, “selama”, “bulan”, “Ramadan”, “mendatang”, “.”], [“Mereka”, “menuntut”, “SK”, “tersebut”, “ditinjau”, “ulang”, “.”]], “extractive_summary”: [1, 5]}	*Text: Liputan6. com, Surabaya: Sekitar 150 pekerja seks komersial (PSK) dari lima lokalisasi yang ada di Jawa Timur, Rabu (9/10) siang, berunjuk rasa di Gedung DPRD Surabaya. Mereka memprotes Surat Keputusan Walikota mengenai penutupan lokalisasi selama bulan Ramadan mendatang. Bahkan surat tersebut sudah diserahkan ke DPRD Surabaya. Para demonstran yang datang dengan dandanan menor itu berasal dari lima lokalisasi, yakni Doli, Jarak, Moroseneng, Tambak Asri, dan Bangun Rejo. Selain para PSK, beberapa pedagang yang menggantungkan hidup di daerah lokalisasi juga turut berunjuk rasa. Dalam orasinya, mereka meminta SK Walikota ditinjau ulang. Menurut mereka, selama bulan puasa mereka harus mengumpulkan uang buat keluarganya untuk keperluan di Hari Raya Idul Fitri. Jika nantinya SK tersebut tetap diberlakukan, mereka berharap instansi terkait konsisten menindak tegas prostitusi kelas atas yang beroperasi di hotel-hotel. (PIN/Benny Christian dan Bambang Ronggo).* *Summary: Sekitar 150 PSK di Surabaya, Jatim, memprotes SK Walikota mengenai penutupan lokalisasi selama bulan Ramadan mendatang. Mereka menuntut SK tersebut ditinjau ulang.*

2.3. Data augmentation

Data augmentation was specifically implemented for the training data in model 3. The data augmentation process utilizes the generative capabilities of the ChatGPT API to produce additional abstractive summaries. The clean news article from approximately 10% of Liputan6’s original training dataset were sent as an input to the ChatGPT API, and instructions to perform abstractive summarization on the batched data was also sent to the ChatGPT API. The data augmentation implementation’s first step was to batch the data from the Liputan6 dataset, which involves grouping the clean news article from multiple files of the dataset into a single file. This strategy was implemented to streamline communication with the ChatGPT API, enabling simultaneous processing of multiple news articles. The batched data were then sent as input to the OpenAI ChatGPT API.

The API utilizes a conversation format, with system instructions providing guidelines for summarization, and user message’s content containing the data to be summarized. The instruction used was “Perform abstractive summarization on the data. Input format will be a json array consisting of text. Output should be in Bahasa Indonesia. Make output length approximately one third or (33%) of input length. Make output format in json array [“item1”, “item2”]. Make sure every item in the output is terminated with, and make the “matches.” The instruction ensured that the generated output are abstractive summaries in Bahasa Indonesia, also ensuring the JSON array structure consistency and preventing syntax errors”. The responses from the ChatGPT API, containing generated abstractive summaries, were then assembled with the original text, and then written to new JSON files. Over 16,700 files were generated, and then combined to the original 10% of the Liputan6 training dataset, resulting in over 36,000 data used to fine-tune the BART model once more. A sample of data augmentation result with ChatGPT can be seen in Table 2.

2.4. Training with BART model

After the data is clean and ready to be feed into the model, the BART model is built. This paper uses BART as the model for abstractive text summarization. BART is a transformer-based model that has both encoder and decoder. BART has a bidirectional encoder and autoregressive decoder. Figure 2 shows the architecture of BART.

The BART architecture that this study uses has an embedding size of 768 with vocabulary size of 50,265, and 6 layers of both encoder and decoder. It has 12 attention heads, feedforward dimension of 3,072 for each encoder and decoder. Padding index for embedding layers is set to 1. Both encoder and decoder have a size of 768 units, with maximum output length of 1,024 tokens. The BART model architecture combines several key components to effectively process and generate text sequences. The input text is tokenized, and the token sequence is then fed to the shared embedding layer of BART, where each token is represented by a hidden state. This shared embedding layer is responsible for creating a meaningful vector representation for each token in the input text. The token embedding then enters the encoder, where each layer employs a feedforward unit consisting of fully connected linear layers. These components expedite the conversion and non-linear processing of input sequences. In addition, the encoder extracts contextual dependencies from the input data by using multi-head attention methods (BartSdpaAttention) [17].

Table 2. Data augmentation

Data Liputan6 after augmentation
{"text": "Liputan6. com, Surabaya: Sekitar 150 pekerja seks komersial (PSK) dari lima lokalisasi yang ada di Jawa Timur, Rabu (9/10) siang, berunjuk rasa di Gedung DPRD Surabaya. Mereka memprotes Surat Keputusan Walikota mengenai penutupan lokalisasi selama bulan Ramadan mendatang. Bahkan surat tersebut sudah diserahkan ke DPRD Surabaya. Para demonstran yang datang dengan dandanan menor itu berasal dari lima lokalisasi, yakni Doli, Jarak, Moroseneng, Tambak Asri, dan Bangun Rejo. Selain para PSK, beberapa pedagang yang menggantungkan hidup di daerah lokalisasi juga turut berunjuk rasa. Dalam orasinya, mereka meminta SK Walikota ditinjau ulang. Menurut mereka, selama bulan puasa mereka harus mengumpulkan uang buat keluarganya untuk keperluan di Hari Raya Idul Fitri. Jika nantinya SK tersebut tetap diberlakukan, mereka berharap instansi terkait konsisten menindak tegas prostitusi kelas atas yang beroperasi di hotel-hotel. (PIN/Benny Christian dan Bambang Ronggo).", "summary": "Sebanyak 150 pekerja seks komersial (PSK) dari lima lokalisasi di Jawa Timur melakukan demo di Gedung DPRD Surabaya pada Rabu (9/10). Mereka memprotes Surat Keputusan Walikota tentang penutupan lokalisasi selama bulan Ramadan. Para demonstran meminta SK Walikota ditinjau ulang karena mereka harus mengumpulkan uang untuk keperluan di Hari Raya Idul Fitri."}

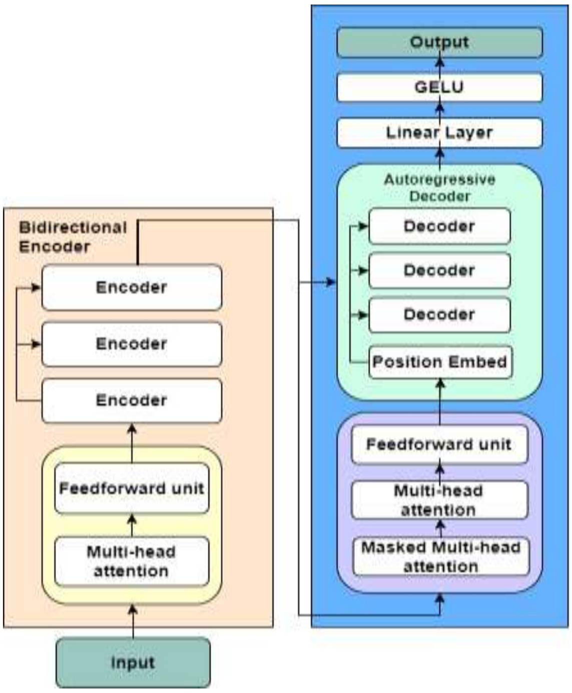


Figure 2. BART architecture

BART also has a special denoising autoencoder that enables us to implement any type of document corruption. This is accomplished via token masking. A document's sentences can be randomly rearranged using sentence permutation. After choosing a token at random from a sentence, document rotation can also be performed so that the token appears at the beginning. A token can also be arbitrarily removed from the original sentence using token deletion. Text infilling can also be done by adding a single mask token to word sequences in place of many tokens or by inserting a mask token at a randomly chosen location [7]. This method is useful in real-world circumstances with noisy or incomplete data because it encourages the model to learn representations that are robust to noise and varied input changes. Sentence permutation and document rotation in particular help the model adjust to various writing styles, document layouts, and linguistic nuances, which improves its output's coherence across a range of text formats.

BART uses an autoregressive method in the decoder section, creating output sequences one token at a time while taking the tokens that were previously generated into account. Every decoder layer has a feedforward unit with linear layers and GELU activation for non-linear transformations. Gaussian error linear unit (GELU), is an activation function that maps input values through a particular mathematical formulation to add non-linearity to the model. The activation function facilitates the identification of intricate patterns and connections within the data [7]. The mathematical expression for the GELU activation function can be approximated as shown in (1).

$$\mathrm{GeLU}(x) = 0.5x\left(1 + \tanh\left(\sqrt{\frac{2}{\pi}}(x + 0.044715x^3)\right)\right) \tag{1}$$

Where x is the input in which the GELU function is applied to, transforming it into a non-linear output, allowing it to learn and represent complex patterns and relationships in the data. 0.5x is a a linear term that allows the GELU function to approach linearity for large positive and negative values of x, and the term inside the hyperbolic tangent function introduces the non-linearity where the cubic term approximates the cumulative distribution function of a Gaussian distribution and the $\sqrt{2/\pi}$ is a constant included to scale the entire expression [18]. Each encoder layer has self-attention mechanism, feedforward neural network, and residual connection that can be represented by (2) and (3).

$$x = GeLU(FC(x + SelfAttn(x)) \tag{2}$$

$$x = x + FC(x) \tag{3}$$

Each decoder layer also has self-attention mechanism, encoder-decoder attention, feedforward neural network, residual connection, and encoder-decoder cross-attention that can be represented by (4)-(7).

$$x = GeLU(FC(x + SelfAttn(x)) \tag{4}$$

$$x = x + WeightAttn\big(CrossAttn(x, encoder_{hidden})\big) \tag{5}$$

$$x = GeLU\big(FC(x)\big) \tag{6}$$

$$x = x + FC(x) \tag{7}$$

Where x represents the input hidden representation in each layer of the model. The stacked equations in both layers operate sequentially, with each equation building upon the output of the previous one. This stacking enables the model to learn hierarchical representations, extracting intricate patterns and relationships from the input data through a series of operations. In the encoder layer, the input hidden embedding is passed through the self-attention mechanism, capturing contextual information, followed by a feedforward neural network and the GELU activation function. The output is then refined through another feedforward operation, incorporating the original input through a residual connection to aid with the flow of information during training and avoids potential vanishing or exploding gradient problems.

In the decoder layer, the initial equation mirrors the encoder’s structure, with self-attention, feedforward operations, and activation functions. The subsequent equations introduce cross-attention to the encoder’s hidden states, allowing the decoder to consider relevant information from the input sequence when generating the output. The input embedding is refined further by implementing feedforward operations, applying GeLU activation function, and adding a residual connection to ensure the information from previous stages is preserved. Following this, the decoder performs cross-attention over the last hidden layer of the encoder and then no additional fully connected layers are utilized for word prediction [19]. The GELU activation function was chosen due to its high performance over other activation functions like ELU and ReLU [20]. It was widely used in not only NLP tasks, but also computer vision tasks in which its performance has been proven to achieve the highest test accuracy of 89.52% on the CIFAR-100 dataset [21].

The decoder also uses masked multi-head attention mechanisms (BartSdpaAttention) in addition to self-attention. To avoid information from future tokens from leaking out and to preserve the autoregressive nature of the generation process, the masked attention makes sure that the model only pays attention to the tokens that come before the current location during decoding [7].

The key, query, and value vector computation are applied to the input embeddings. These are used to compute attention weights through a dot product operation between the key and query vectors. The attention score then goes through GELU activation function, and the output will be multiplied by the value vector. The feedforward layer receives this attention output subsequently. Notably, BART uses multi-head attention, which involves several parallel computations of the self-attention, just like other transformers model. This allows the model to capture diverse aspects and interactions inside the input sequence simultaneously, boosting its ability to understand complex dependencies during both encoding and decoding phases [22].

2.5. Evaluation with ROUGE score

The last step is to evaluate the model's performance. This study uses ROUGE as its evaluation metric. ROUGE compares a machine-generated summary with the gold summary that was created by a human, and it is often used to evaluate the quality of text summarization algorithm's summary result. The Liputan6 dataset provides abstractive gold summary in all of its dataset variants, including the test set. The test set of Liputan6 is used as the evaluation dataset within the trainer configuration. This means that during the training process, model performance and generalization are assessed on this specific test set. It serves as a benchmark to evaluate how well the model has learned from the training data and to estimate its performance in generating abstractive summaries when applied to new data from the Liputan6 test set.

There are some methods for calculating ROUGE that were based on the difference of granularity, such as ROUGE-N, ROUGE-L, ROUGE-S, ROUGE-SU. Between those methods, ROUGE-1, ROUGE-2, and ROUGE-L are the most used metrics in literature cause reflect the granularity of the studied texts. ROUGE-1 and ROUGE-2 are the part of ROUGE-N where ROUGE-N refers to the overlapping of N-grams between the machine-generated summary and the golden summary. ROUGE-L measures the longest common word sequence that was computed by the longest common subsequence algorithm [23]. This work will use ROUGE-1, ROUGE-2, and ROUGE-L for the evaluation.

Aside from evaluating using ROUGE score, human inference on the generated summary will also be conducted to make sure the summary is contextually correct and human-friendly. The inference is done by encoding the input text using a tokenizer and checking its token length against a specified maximum length. If the text fits within this limit, the model generates a summary without any further processing. For longer texts exceeding the limit, the model splits the text into manageable chunks and recursively summarizes each chunk. This approach allows the function to handle texts of varying lengths while adhering to the token limit, ensuring efficient and quality summarization.

3. RESULTS AND DISCUSSION

In the realm of Indonesian text summarization, there exists a significant gap in understanding the effects of multiple datasets fine-tuning and text augmentation on performance. While earlier studies have delved into the impact of fine-tuning on model performance and the benefits of text augmentation using ChatGPT, they have not explicitly addressed how these techniques influence abstractive text summarization in Indonesian language.

The implementation and experimentation for this study were conducted using NVIDIA GeForce RTX 4060 Ti GPU, which provides dedicated processing unit to accelerate model training and experimentation. The datasets that were used to train the model are open-source Indonesian text summarization datasets, Indosum and Liputan6. The usage of multiple datasets introduces diversity to the model, so that it can learn to generalize its understanding across different contexts and linguistic variations present in the training data. The hyperparameter used as the trainer's configuration are listed in Table 3.

For the first model fine-tuning with Indosum dataset, a relatively low learning rate was chosen because the Indosum dataset is less abstractive. The nature of Indosum dataset who has less abstractive summarization may contain noise in the form of sentences that are not necessarily the most informative for summarization. The lower learning rate allowed the model to fine-tune its parameters more gradually to better adapt to the nuances of less abstractive summaries and the inherent noise that comes with the less abstractive summarization data [24]. Since Indosum was also used to expand the vocabulary of the model, the training hyperparameters were set to strike a balance with smaller learning rate and more epochs to allow the model to learn new Indonesian vocabulary. This is crucial for the model to handle a broader range of words and expressions when later fine-tuning on more abstractive data. A smaller batch size was also used to introduce more variance in the updates, to help the model generalize better to less abstractive nature of Indosum dataset [25].

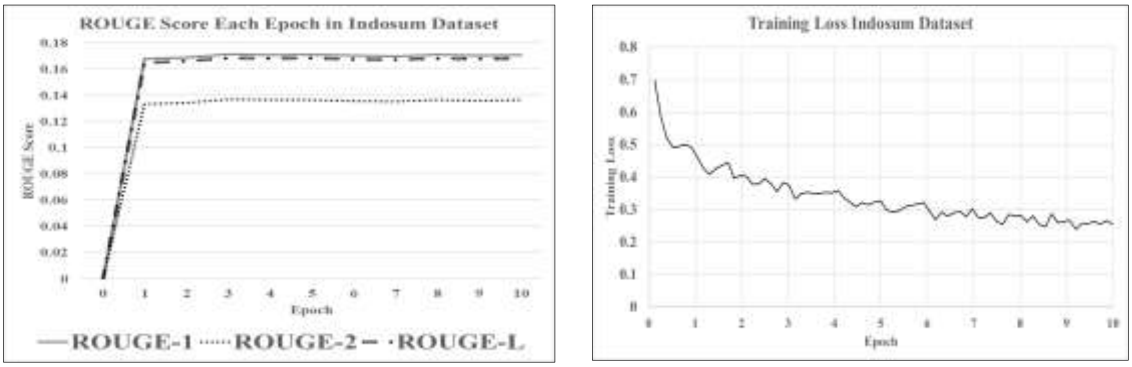
A slightly larger, but still relatively low learning rate for the second fine-tuning of the model for the downstream task was chosen for the larger and abstractive Liputan6 dataset to allow for more substantial updates to model parameters based on the new data. A larger batch size was also used to efficiently use computational resources when working with a larger dataset, allowing for faster training. Fewer epochs were also chosen to help prevent overfitting. The learning rate and number of epochs are carefully tuned to strike a balance, aiming for optimal model convergence, and improving the model's generalization [26].

A very low learning rate was chosen for fine-tuning with augmented data. The goal is to make small adjustments based on the augmented data without overriding the knowledge gained from the previous datasets. The same batch size as the previous fine-tuning step was used to balance computational efficiency and effective learning. Training for more epochs allows the model to distinguish between meaningful patterns and noise that might be generated by external models like ChatGPT, helping it focus on learning the important information from the augmented examples. A small learning rate combined with more epochs allowed the model to make careful adjustments over extended training period, which led to improved model performance and generalization. A weight decay of 0.01 was also implemented to prevent overfitting by penalizing large weights, encouraging it to use smaller weights for better generalization on new data [24]. The chosen learning rate aligns with the findings of a previous study in the natural language processing field using transformer-based model, which demonstrated the efficacy of values within the range of 5e-6 to 1e-4 [27]. After the multiple datasets fine-tuning process was completed, the model's performance is evaluated using ROUGE metrics to assess the quality of the generated abstractive summaries. Aside from ROUGE metrics, training loss graphs are also included to demonstrate how well the model is learning over the course of the training process. The training loss and ROUGE scores graphs are shown in Figure 3 for Indosum dataset, Figure 4 for Liputan6 dataset, and Figure 5 for Liputan6 augmented dataset.

Figures 3(a), 4(a), and 5(a) show escalation in overall ROUGE score as the epoch progresses. Figures 3(b), 4(b), and 5(b) also shows that the relationship between number of epochs and training loss is inversely proportional. It indicates that the difference between the predicted output of the model and the actual target values in the training dataset is getting smaller as the epoch progresses [28]. The downward trend in the loss curve demonstrates that the model is progressively adapting to the complexity of the underlying patterns, showing its ability to extract important information from the news articles and generating coherent summaries. The ROUGE scores comparison with previous studies are also included and can be seen in Table 4.

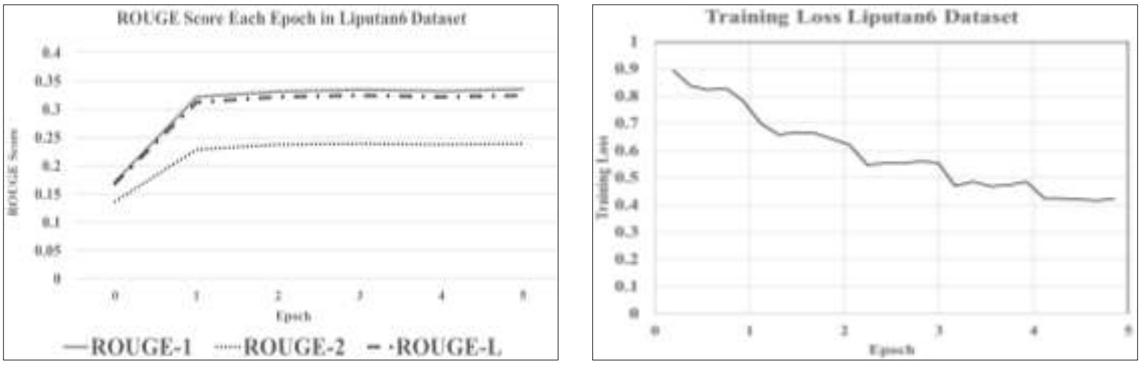
Table 3. Hyperparameter configuration

Dataset	Learning rate	Epoch	Batch size	Weight decay
Indosum	2e-5	10	3	0.01
Liputan6	5e-5	5	8	0.01
Liputan6 augmented	10e-6	10	8	0.01



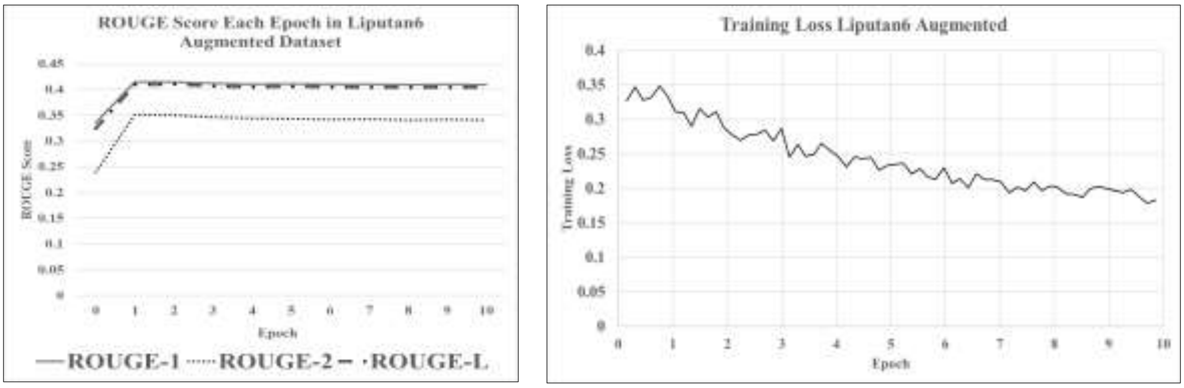
(a) (b)

Figure 3. The training loss and ROUGE scores graphs for Indosum dataset (a) ROUGE score and (b) training loss



(a) (b)

Figure 4. The training loss and ROUGE scores graphs for Liputan6 dataset (a) ROUGE score and (b) training loss



(a) (b)

Figure 5. The training loss and ROUGE scores graphs for Liputan6 augmented dataset (a) ROUGE score and (b) training loss

Table 4. Table comparison with previous research

	Model	Dataset	R1	R2	RL
	LEAD-1	Liputan6 Canonical	32.67	18.50	29.40
	LEAD-2	Liputan6 Canonical	36.68	20.23	33.71
	LEAD-3	Liputan6 Canonical	34.49	18.84	32.06
	ORACLE	Liputan6 Canonical	51.54	30.56	47.75
	PTGEN	Liputan6 Canonical	36.10	19.19	33.56
	PTGEN+COV	Liputan6 Canonical	35.53	18.56	32.92
	BERTEXT (mBERT)	Liputan6 Canonical	37.51	20.15	34.57
	BERTABS (mBERT)	Liputan6 Canonical	39.48	21.59	36.72
	BERTEXTABS (mBERT)	Liputan6 Canonical	39.81	21.84	37.02
	BERTEXT (IndoBERT)	Liputan6 Canonical	38.03	20.72	35.07
	BERTABS (IndoBERT)	Liputan6 Canonical	40.94	23.01	37.89
	BERTEXTABS (IndoBERT)	Liputan6 Canonical	41.08	22.85	38.01
Research result	Model I BART-Indosum	Indosum	17.08	13.65	16.82
	Model II BART-Indosum-Liputan6	Indosum, Liputan6 Canonical	33.51	23.91	32.47
	Model III BART-Indosum-Liputan6-Liputan6Augmented	Indosum, Liputan6 Canonical, Liputan6 Augmented GPT3.5	40.93	34.09	40.37

From Table 4, compelling results are shown regarding the effects of multiple dataset fine-tuning and text augmentation on Indonesian text summarization ROUGE score. It is found that multiple fine-tuning iterations and improving training data with text augmentation correlated positively with improved ROUGE scores. Furthermore, the proposed method in this study tended to produce summaries with a higher proportion of relevant and coherent information as measured by ROUGE metrics. The lower ROUGE scores following fine-tuning with the Indosum dataset can be rationalized by the fact that the Indosum summaries are inherently less abstractive, and the model may not have been exposed to enough Indonesian text to sufficiently capture the nuances of Indonesian semantics and vocabulary. As the model was fine-tuned with Liputan6 dataset which are more abstractive in nature, the model becomes increasingly adept at capturing and reproducing abstractive elements. The study suggests that multiple datasets fine-tuning and higher training data quality with help of text augmentation is associated with improved text summarization quality as measured by ROUGE metrics. Comparing the study results with other paper, our study aligns well with prior research in [11], where the paper demonstrates that fine-tuning with multiple datasets increased the ROUGE-1 score of the BART model. It was shown that fine-tuning solely with CNN dataset yields a ROUGE-1 score of 0.1570, while incorporating both CNN and SAMSum dataset results in an increased ROUGE-1 score of 0.1834. The subsequent fine-tuning with Liputan6 augmented dataset proves to be able to enhance the model’s ability to generate summaries that encapsulate nuanced Indonesian semantics. This augmentation contributes to a further increase in ROUGE scores, emphasizing the positive impact of introducing abstractive variations in the training data, as also demonstrated in paper [29].

From Table 4, it is also shown that BART model that was finetuned using Indosum, Liputan6, and augmented Liputan6 dataset overperformed most of the previous research’s performance based on its ROUGE score, with the exception of the ORACLE, BERTABS(IndoBERT), and BERTEXTABS(IndoBERT) model. BART model that was finetuned using Indosum, Liputan6, and augmented Liputan6 dataset has the best ROUGE-2 score, outperforming ORACLE’s model although

ORACLE still has the best ROUGE-1 and ROUGE-L score. The model is proven to have competitive performance, even with computation limitations, leaving room for improvement through further hyperparameters fine-tuning and other optimization techniques. In addition to ROUGE metrics, the generated summaries produced by the model will also be shown as an alternative evaluation measure, ensuring that the model's generated summary aligns with human-friendly standards. Comparing with other paper, the previous study [12] using BERT reported issues such as incomplete sentences, incorrect phrases, and repeated words in the generated summaries. Additionally, BertSum was found to produce a number of meaningless words due to sub-word tokenization. In contrast, this study's approach utilizing BART for fine-tuning demonstrated a notable improvement in summary quality. This method generated summaries that were more coherent, accurate, and devoid of the issues commonly associated with BERT-based summarization. Specifically, the model generated fewer incomplete sentences, more accurate phrasing, and a reduction in repeated and meaningless words. A sample of the model's generated summary is shown in Table 5.

Table 5. Summary result

	Summary
Original article 1	*Liputan6. com, Bogor: Dua tersangka pemerkosa anak di bawah umur, Basir, dan Masum, baru-baru ini, dibekuk petugas Kepolisian Resor Kota Bogor, Jawa Barat. Di hadapan polisi, Basir mengaku melakukan tindakan asusila dan membunuh bocah itu setelah menonton video cassette disc porno di rumah kakaknya. Untuk menghilangkan jejak, Basir yang sehari-hari pengangguran itu secara sadis membunuh korban dengan cara mencekik dan menggantungnya di pohon. Tiga hari kemudian, Tim Buru Sergap Polres Bogor juga menangkap pelaku lain, Masum. Hingga kini kedua tersangka ditahan di Mapolres Bogor. Kedua orang itu akan diperiksa seorang psikiater. (COK/Gaby Getal dan Muhamad Guntur).*
Gold summary	*Dua tersangka pemerkosa anak di Bogor ditangkap. Para pelaku mengaku menggagahi bocah di bawah umur setelah melihat tayangan porno.*
Liputan6 augmented model summary	*Dua tersangka pemerkosa anak di Bogor berhasil ditangkap setelah menonton video cassette disc porno di rumah kakaknya. Mereka membunuh korban dengan cara mencekik dan menggantungnya di pohon.*
Liputan6 model summary	*Dua tersangka pemerkosa anak di bawah umur dibekuk karena melakukan tindakan asusila dan membunuh bocah itu setelah menonton video cassette disc porno di rumah kakaknya.*

In the Liputan6 augmented model summary, it can be seen that the model is able to form new words, such as "*berhasil*" which means successful, and then "*ditangkap*" which replaces "dibekuk", both referring to "arrest". The model is also able to combine the essence of two sentences together ("*Dua tersangka pemerkosa anak di bawah umur, Basir, dan Masum, baru-baru ini, dibekuk petugas Kepolisian Resor Kota Bogor, Jawa Barat.*" and "*Di hadapan polisi, Basir mengaku melakukan tindakan asusila dan membunuh bocah itu setelah menonton video cassette disc porno di rumah kakaknya*.") into one concise sentence. The model can also substitute the phrase "*dua tersangka pemerkosa anak*" (which means two suspected child rapists) with the pronoun "*mereka*" (meaning "they"). It also provides more detail of the incident by detailing how the victim was murdered by the two criminals. The model is able to capture the essence of the news article, seamlessly integrating new words into the summary and replacing words with its synonyms into the summary without disrupting the context or making any misleading information. The Liputan6 model summarizes news articles concisely and abstractively, capturing key sentences and generating cause-and-effect sentences such as "*karena melakukan tindakan asusila dan membunuh bocah itu*" as the reason why "*Dua tersangka pemerkosa anak di bawah umur dibekuk*." From the sample, the Liputan6 augmented model outperforms the Liputan6 model as it exhibits a capability to generate new words, enhancing its abstractive nature.

Despite the promising results, this study has limitations due to the use of only 10% of the available training data because of computational resource constraints which may have affected the model's ability to generalize and achieve optimal performance. Additionally, the text augmentation process using ChatGPT involved batching multiple news articles and truncation due to the API's token limit, potentially leading to the loss of important contextual information and impacting the quality of augmented data. Future research may explore the use of larger training datasets and alternative text augmentation techniques to overcome the limitations observed in this study. Additionally, investigating the impact of fine-tuning on other languages and summarization tasks could further enrich people's understanding of these techniques' generalizability and effectiveness. In conclusion, this study provides valuable insights into improving Indonesian text summarization through multiple datasets fine-tuning and text augmentation. This study suggest that the approach can lead to improved ROUGE scores without compromising the quality of the summaries. While there are limitations to consider, the findings from this study offer a promising direction for future research in the field of abstractive text summarization.

4. CONCLUSION

Based on the ROUGE evaluation score, fine-tuning using Indosum, Liputan6, and Liputan6 augmented datasets increased the performance of the BART model in performing abstractive Indonesian text summarization. Data augmentation using ChatGPT for text summarization also contributes increasing the ROUGE score and performance of the summarization model and generates more diverse sentence structures and new vocabulary that are not present in the original article, helping the model to learn Indonesian semantic and language nuances further. This study also shows that the novel approach of fine-tuning with multiple datasets and integrating data augmentation using ChatGPT outperforms some of the previous research based on its ROUGE scores. For future research, this type of approach can be implemented in different types of models and different datasets can also be used to fine-tune the model. Measuring the performance by its news category, cross-lingual summarization using this approach, fine-tuning hyperparameter optimization, incremental fine-tuning that allows the model to adapt to new data continuously, different approaches of augmentation such as backtranslation, and ethical considerations related to the use of an abstract summarization model, including potential biases and fairness issues, can be opportunities for further research.

REFERENCES

[1] A. V. Das and M. Viswanathan, "Ownership, privacy, and value of health-care data: Perspectives and future direction," *IHOPE Journal of Ophthalmology*, vol. 2, pp. 41–46, May 2023, doi: 10.25259/ihopejo_33_2022.

[2] A. S. Girsang and F. J. Amadeus, "Extractive text summarization for indonesian news article using ant system algorithm," *Journal of Advances in Information Technology*, vol. 14, no. 2, pp. 295–301, 2023, doi: 10.12720/jait.14.2.295-301.

[3] V. Risne and A. Siitova, "Text summarization using transfer learning extractive and abstractive summarization using BERT and GPT-2 on news and podcast data," University of Gothenburg, 2019.

[4] P. K. Amaravarapu and A. Khare, "Abstractive text summarization with neural network based sequence-to-sequence models," Politecnico di Milano, 2020.

[5] F. Koto, T. Baldwin, and J. H. Lau, "FFCI: a framework for interpretable automatic evaluation of summarization," *Journal of Artificial Intelligence Research*, vol. 73, pp. 1553–1607, Apr. 2022, doi: 10.1613/jair.1.13167.

[6] D. Liu *et al.*, "GLGE: a new general language generation evaluation benchmark," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 408–420, doi: 10.18653/v1/2021.findings-acl.36.

[7] M. Lewis *et al.*, "BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7871–7880, doi: 10.18653/v1/2020.acl-main.703.

[8] F. Koto, J. H. Lau, and T. Baldwin, "Liputan6: a large-scale indonesian dataset for text summarization," *arxiv*, 2020, [Online]. Available: http://arxiv.org/abs/2011.00679.

[9] S. Chhabra, P. Majumdar, M. Vatsa, and R. Singh, "Data fine-tuning," *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, vol. 33, no. 01, pp. 8223–8230, Jul. 2019, doi: 10.1609/aaai.v33i01.33018223.

[10] A. Arnab *et al.*, "Beyond transfer learning: co-finetuning for action localisation," *arxiv*, 2022, [Online]. Available: http://arxiv.org/abs/2207.03807.

[11] N. Giarelis, C. Mastrokostas, and N. Karacapilidis, "Abstractive vs. extractive summarization: an experimental review," *Applied Sciences (Switzerland)*, vol. 13, no. 13, p. 7620, Jun. 2023, doi: 10.3390/app13137620.

[12] A. Fabbri *et al.*, "Improving zero and few-shot abstractive summarization with intermediate fine-tuning and data augmentation," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 704–717, doi: 10.18653/v1/2021.naacl-main.57.

[13] Y. Fang, X. Li, S. W. Thomas, and X. Zhu, "ChatGPT as data augmentation for compositional generalization: a case study in open intent detection," *FinNLP-Muffin 2023 - Joint Workshop of the 5th Financial Technology and Natural Language Processing and 2nd Multimodal AI For Financial Forecasting, in conjunction with IJCAI 2023 - Proceedings*, pp. 13–33, 2023.

[14] Z. Jiang, M. Wang, X. Cai, D. Gao, and L. Yang, "ChatGPT based contrastive learning for radiology report summarization," *SSRN*, 2023, doi: 10.2139/ssrn.4485806.

[15] S. Cahyawijaya *et al.*, "IndoNLG: benchmark and resources for evaluating indonesian natural language generation," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 8875–8898, doi: 10.18653/v1/2021.emnlp-main.699.

[16] R. Wijayanti, M. L. Khodra, and D. H. Widyantoro, "Indonesian abstractive summarization using pre-trained model," in *3rd 2021 East Indonesia Conference on Computer and Information Technology, EIConCIT 2021*, Apr. 2021, pp. 79–84, doi: 10.1109/EIConCIT50028.2021.9431880.

[17] S. Li, H. Yan, and X. Qiu, "Contrast and generation make BART a good dialogue emotion recognizer," *Proceedings of the 36th AAAI Conference on Artificial Intelligence, AAAI 2022*, vol. 36, no. 10, pp. 11002–11010, Jun. 2022, doi: 10.1609/aaai.v36i10.21348.

[18] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," *arxiv*, 2016, [Online]. Available: http://arxiv.org/abs/1606.08415.

[19] T. Chen, X. Wang, T. Yue, X. Bai, C. X. Le, and W. Wang, "Enhancing abstractive summarization with extracted knowledge graphs and multi-source transformers," *Applied Sciences (Switzerland)*, vol. 13, no. 13, p. 7753, Jun. 2023, doi: 10.3390/app13137753.

[20] N. Hayatin, K. M. Ghufron, and G. W. Wicaksono, "Summarization of COVID-19 news documents deep learning-based using transformer architecture," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 19, no. 3, pp. 754–761, Jun. 2021, doi: 10.12928/TELKOMNIKA.v19i3.18356.

[21] M. Lee, "Mathematical analysis and performance evaluation of the GELU activation function in deep learning," *Journal of Mathematics*, vol. 2023, pp. 1–13, Aug. 2023, doi: 10.1155/2023/4229924.

[22] A. A. Syed, F. L. Gaol, and T. Matsuo, "A survey of the state-of-the-art models in neural abstractive text summarization," *IEEE Access*, vol. 9, pp. 13248–13265, 2021, doi: 10.1109/ACCESS.2021.3052783.

[23] M. Barbella and G. Tortora, "Rouge metric evaluation for text summarization techniques," *SSRN Electronic Journal*, 2022, doi: 10.2139/ssrn.4120317.

[24] K. You, M. Long, J. Wang, and M. I. Jordan, "How does learning rate decay help modern neural networks?," *arxiv*, Aug. 2019, [Online]. Available: http://arxiv.org/abs/1908.01878.

[25] À. R. Atrio and A. Popescu-Belis, "Small batch sizes improve training of low-resource neural MT," Mar. 2022, [Online]. Available: http://arxiv.org/abs/2203.10579.

[26] Y. Li, C. Wei, and T. Ma, "Towards explaining the regularization effect of initial large learning rate in training neural networks," *Advances in Neural Information Processing Systems*, vol. 32, Jul. 2019, [Online]. Available: http://arxiv.org/abs/1907.04595.

[27] S. Casola, I. Lauriola, and A. Lavelli, "Pre-trained transformers: an empirical comparison," *Machine Learning with Applications*, vol. 9, p. 100334, Sep. 2022, doi: 10.1016/j.mlwa.2022.100334.

[28] D. Yoo and I. S. Kweon, "Learning loss for active learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2019, vol. 2019-June, pp. 93–102, doi: 10.1109/CVPR.2019.00018.

[29] H. Dai *et al.*, "AugGPT: leveraging ChatGPT for text data augmentation," *arxiv*, 2023, [Online]. Available: http://arxiv.org/abs/2302.13007.

BIOGRAPHIES OF AUTHORS

Miracle Aurelia will received her bachelor degree at Bina Nusantara University for Computer Science in 2024. Now, she is pursuing her Master Degree at Bina Nusantara University for Computer Science Program. She can be contacted at email: miracle.aurelia@binus.ac.id.

Sheila Monica will received her bachelor degree at Bina Nusantara University for Computer Science in 2024. Now, she is pursuing her Master Degree at Bina Nusantara University for Computer Science Program. She can be contacted at email: sheila.monica@binus.ac.id.

Abba Suganda Girsang is currently a lecturer at master in Computer Science, Bina Nusantara University, Jakarta, Indonesia Since 2015. He got Ph.D. degree in 2015 at the Institute of Computer and Communication Engineering, Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan. He graduated bachelor from the Department of Electrical Engineering, Gadjah Mada University (UGM), Yogyakarta, Indonesia, in 2000. He then continued his master's degree in the Department of Computer Science at the same university in 2006-2008. He was a staff consultant programmer in Bethesda Hospital, Yogyakarta, in 2001 and also worked as a web developer in 2002-2003. He then joined the faculty of the Department of Informatics Engineering in Janabadra University as a lecturer in 2003-2015. His research interests include swarm, intelligence, combinatorial optimization, and decision support system. He can be contacted at email: agirsang@binus.edu.